

2003 DEC 2005

PCT/JP03/13559

日本国特許庁  
JAPAN PATENT OFFICE

23.10.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2002年10月25日

出願番号  
Application Number: 特願2002-310618  
[ST. 10/C]: [JP2002-310618]

RECEIVED	
12 DEC 2003	
WIPO	PCT

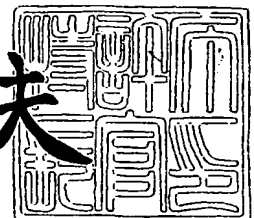
出願人  
Applicant(s): 株式会社ターボデータラボラトリー

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2003年11月27日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 PK020043

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/30

【発明者】

【住所又は居所】 神奈川県横浜市神奈川区松見町四丁目 1 1 0 1 番地 7 コ  
ートハウス菊名 8 0 4 号

【氏名】 古庄 晋二

【特許出願人】

【識別番号】 502369012

【氏名又は名称】 株式会社ターボデータラボラトリー

【代理人】

【識別番号】 100103632

【弁理士】

【氏名又は名称】 窪田 英一郎

【選任した代理人】

【識別番号】 100099715

【弁理士】

【氏名又は名称】 吉田 聡

【手数料の表示】

【予納台帳番号】 058377

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理方法およびデータ処理プログラム

【特許請求の範囲】

【請求項 1】 各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するためのデータ処理方法であって、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、

d) 前記項目値番号配列中の値を参照して、位置の要素を格納した第 2 のポインタ配列および値の要素を格納した第 2 の値リストを生成するステップとを備え、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第 2 のポインタ配列中の要素、および、第 2 のポインタ配列中の要素が指し示す位置の前記第 2 の値リスト中の要素を介して、値リスト中の値が特定されることを特徴とするデータ処理方法。

【請求項 2】 前記第 2 のポインタ配列および第 2 の値リストを生成するステップが、

前記項目値番号配列中の要素をソートし、かつ、重複値が排除された状態となった配列を生成し、これを第 2 の値リストとするステップと、

前記ソートおよび重複値の排除を反映するように、前記項目値番号配列中の要素を変換して、第 2 のポインタ配列を生成するステップとを有することを特徴とする請求項 1 に記載のデータ処理方法。

【請求項 3】 各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するためのデータ処理方法であって、

- a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、
- b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、
- c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、
- e) 前記項目値番号配列中を第 2 の値リストとして利用するとともに、もとの値リストを第 2 の値リストとして利用するステップとを備え、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第 2 のポインタ配列中の要素、および、第 2 のポインタ配列中の要素が指し示す位置の前記第 2 の値リスト中の要素を介して、値リスト中の値が特定されることを特徴とするデータ処理方法。

【請求項 4】 さらに、前記順序付き集合配列のサイズと、前記値リストのサイズとを比較するステップを備え、

前記比較の結果、部分集合のサイズが、値リストのサイズより所定の割合で小さい場合に、前記 c) および d)、または、前記 c) および e) のステップを実行することを特徴とする請求項 1 ないし 3 の何れか一項に記載のデータ処理方法。

【請求項 5】 請求項 1 ないし 4 の何れか一項に記載のデータ処理方法により生成された第 2 のポインタ配列および／または第 2 の値リストを利用した検索方法であって、

第 2 の値リスト中の要素のうち、検索対象となる項目値を指し示す要素を特定

するステップと、

前記第2の値リストと同じサイズのフラグ配列中、前記特定された要素の位置と対応する位置に、フラグがオンである状態を示す値を配置するステップと、

前記順序付き集合中のレコード番号が指し示す、第2のポインタ配列の要素を特定するステップと、

前記フラグ配列中、前記第2のポインタ配列の要素が指し示す位置のフラグの状態を参照するステップと、

前記フラグの状態がオンである場合、前記レコード番号を、新たに設けられた出力用の順序付き集合配列に、順次配置するステップとを備えたことを特徴とする方法。

【請求項6】 請求項1ないし4の何れか一項に記載のデータ処理方法により生成された第2のポインタ配列および／または第2の値リストを利用した集計方法であって、

前記第2の値リストの要素に対応して、値のカテゴリを示す分類番号を配置する分類番号配列を生成するステップと、

指定された分類番号に対応する、前記順序付き集合配列中のレコード番号を特定するステップと、

前記特定されたレコード番号が指し示す、所定の値リストの項目値を用いた集計を実行するステップとを備えたことを特徴とする方法。

【請求項7】 請求項1ないし4の何れか一項に記載のデータ処理方法により生成された第2のポインタ配列および／または第2の値リストを利用したソート方法であって、

前記第2の値リストの値ごとに、前記第2のポインタ配列の指し示す要素の数である存在数を算出するステップと、

前記存在数に基づき、前記第2の値リストの値に対応する、前記順序付き集合配列中のレコード番号を配置すべき先頭の位置を示す累計数配列を生成するステップと、

前記累計数配列を参照して、前記値リスト中の項目値のソート順が反映されるように、前記順序付き集合配列のレコード番号を、出力用の配列中に配置するス

テップとを備えたことを特徴とする方法。

【請求項 8】 請求項 1 ないし 4 の何れか一項に記載のデータ処理方法により、生成された、複数の表形式データのそれぞれに関する情報ブロックにおける、第 2 のポインタ配列および／または第 2 の値リストを利用して、表形式データを結合する方法であって、

複数の表形式データのそれぞれにおいて、共有化すべき項目を見出すステップと、

当該項目に関する情報ブロックの第 2 の値リスト中の項目値を等価にするステップと、

前記項目値を等価にすることに応答して、それぞれの情報ブロックにおいて、第 2 のポインタ配列中の要素を、前記項目値の配置の変化に合致するように更新するステップとを備えたことを特徴とする方法。

【請求項 9】 各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するために、コンピュータを動作させるデータ処理プログラムであって、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、

d) 前記項目値番号配列中の値を参照して、位置の要素を格納した第 2 のポインタ配列および値の要素を格納した第 2 の値リストを生成するステップとを、前記コンピュータに実行させ、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置

の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定されるように、前記コンピュータを動作させることを特徴とするデータ処理プログラム。

【請求項10】 前記第2のポインタ配列および第2の値リストを生成するステップにおいて、

前記項目値番号配列中の要素をソートし、かつ、重複値が排除された状態となった配列を生成し、これを第2の値リストとするステップと、

前記ソートおよび重複値の排除を反映するように、前記項目値番号配列中の要素を変換して、第2のポインタ配列を生成するステップとを前記コンピュータに実行させることを特徴とする請求項9に記載のデータ処理プログラム。

【請求項11】 各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するために、コンピュータを動作させるデータ処理プログラムであって、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、

e) 前記項目値番号配列中を第2の値リストとして利用するとともに、もとの値リストを第2の値リストとして利用するステップとを、前記コンピュータに実行させ、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定されるよ

うに、前記コンピュータを動作させることを特徴とするデータ処理プログラム。

【請求項 12】 さらに、前記順序付き集合配列のサイズと、前記値リストのサイズとを比較するステップを、前記コンピュータに実行させ、

前記比較の結果、部分集合のサイズが、値リストのサイズより所定の割合で小さい場合に、前記 c) および d)、または、前記 c) および e) のステップを、前記コンピュータに実行させることを特徴とする請求項 9 ないし 11 の何れか一項に記載のデータ処理プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、表形式データの部分集合を効率的に処理するための方法およびプログラムに関する。

【0002】

【従来の技術】

データベースなど大量のデータを、検索、集計、ソートするため、或いは、これらのデータをジョイン、更新等するためには多大な時間を要する。これを解決すべく、本発明者は、表形式のデータを極めて拘束に検索、集計およびソートするための手法、並びに、表形式のデータをジョインし、更新し、或いは、トランザクション処理する手法を提案している（たとえば、特許文献 1～3 参照）。

【0003】

【特許文献 1】

国際公開第 WO00/10103 号パンフレット

【特許文献 2】

特開 2000-339390 号公報

【特許文献 3】

特開 2001-043290 号公報

【0004】

これら文献に提示された一連の方法は、一切のインデックスを用いず、統一的な処理を実行することができ、かつ、部分集合に対しても効率を落とすことなく



処理を実行することが可能であるという点で革新的である。

本発明者の考案にかかる表形式データは、以下に説明するようなデータ構造を備えている。図1(a)は、表形式データの例を示す図である。この例では、レコード番号を格納した配列100中の各レコード番号のそれぞれに、「会員名」、「地域」および「ファン」という項目の値(項目値)が対応付けられている(符号101参照)。本発明者の考案にかかるデータ構造は、図1(a)に示すような表形式データを、図2(a)～(c)に示すような形式で保持している。

#### 【0005】

たとえば、図2(a)に示すように、項目「会員名」に関して、当該項目値が50温順(英字ではアルファベット順)でソートされた値リスト(以下、場合によって「VL」と称する。)211と、各レコード番号に対応して、当該レコード番号が指し示す値リスト211中の要素位置を示す番号が格納された、値リストへのポインタ配列(以下、場合によって「PV」と称する)212とが設けられている。このようにある項目に関して、ポインタ配列および値リスト、および、これらに加えて場合によって付随する一以上の配列を含む配列群を、情報ブロック(符号201参照)と称する。図2(a)から理解できるように、ポインタ配列は、表形式データのレコード数と同じ数の要素を含む。項目「地域」に関しても、情報ブロック202は、PVおよびVLを含み、同様に、項目「ファン」に関しても、情報ブロック203は、PVおよびVLを含む。

#### 【0006】

レコード番号「0」に着目すると、情報ブロック201中のPVの値は、「5」であるから、格納位置が「5」であるようなVL中の項目値「Green e」が特定される。同様に、情報ブロック202、203のPVの値は、それぞれ、「5」および「2」であるため、対応する格納位置番号の項目値「Tokyo」および「G-T eam」が特定される。これらは、図1(a)におけるレコード番号「1」に含まれるそれぞれの項目値と一致することが理解できるであろう。

このようなデータ構造を利用して、検索、集計、ソート、さらには、ジョインをすることにより、著しく高速な処理を実現している。

#### 【0007】

**【発明が解決しようとする課題】**

しかしながら、上記データ構造を用いた検索等の処理において、以下のような問題点が生じることが認識された。

全体のレコード数から少数の部分集合を処理対象として、部分集合中から所望のレコードを検索する場合であっても、本出願人が考案した手法においては、Vしと同じサイズの配列を設け、当該配列の要素として、検索される値の範囲に含まれるか否かを示すフラグを配置する。同様に、検索の合否判定にも値リストと同じサイズの配列が必要となる。

**【0008】**

ソート時や集計時においても、値リストと同じサイズの一以上の配列を要する。さらに、ジョインの処理時にも、値リストと同じサイズの種々の配列を設ける必要がある。

したがって、著しく大きなレコード数から、ごく少数の部分集合を処理対象として検索、集計、ソートおよびジョインの処理を行う場合であっても、情報ブロック中に大規模な値リストが設けられていると、処理に必要な配列のために大きなメモリ領域を必要とする。また、当該配列中に値を配置する処理、値を走査する処理などを要するため、レコード全体を処理対象とした場合と比較して、処理対象が少数である割には、処理時間を短縮することができないという問題点があった。

**【0009】**

本発明は、上記配列を部分集合の大きさ、これに伴って限定される値リストの部分集合の大きさに合致させて、部分集合の大きさに相当する処理時間の短縮を実現する手法を提供することを目的とする。

**【0010】****【課題を解決するための手段】**

本発明の目的は、各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するためのデータ処理方法であって、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値

番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、

d) 前記項目値番号配列中の値を参照して、位置の要素を格納した第2のポインタ配列および値の要素を格納した第2の値リストを生成するステップとを備え、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定されることを特徴とするデータ処理方法により達成される。

#### 【0011】

本発明によれば、部分集合を構成するレコード番号に関連する値リストの項目値を指し示す第2の値リストを生成し、第2の値リストを参照して、データ処理（後述するような、検索、集計、ソート、結合処理）を実行することが可能となる。したがって、参照するデータ量を小さくできるため処理時間を短縮化することができる。また、データ処理の際に生成される配列のサイズも、第2の値リストや第2のポインタ配列と同じにすることができるため、必要とされるメモリ領域を小さくすることが可能となる。

#### 【0012】

好ましい実施態様においては、前記第2のポインタ配列および第2の値リストを生成するステップが、前記項目値番号配列中の要素をソートし、かつ、重複値が排除された状態となった配列を生成し、これを第2の値リストとするステップと、前記ソートおよび重複値の排除を反映するように、前記項目値番号配列中の要素を変換して、第2のポインタ配列を生成するステップとを有する。

**【0013】**

また、別の好ましい実施態様において、各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するためのデータ処理方法は、

- a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、
- b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、
- c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置するステップと、
- e) 前記項目値番号配列中を第2の値リストとして利用するとともに、もとの値リストを第2の値リストとして利用するステップとを備え、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定される。

**【0014】**

この実施態様においては、第2の値リストは、もとの値リストであるため、そのサイズを小さくすることはできないが、第2のポインタ配列として、順序付き集合配列と同じサイズの項目値番号配列を利用している。したがって、もとのポインタ配列およびもとの値リストを利用する場合と比較すると、参照するデータ量を小さくでき、処理時間を短縮化することが可能となる。

**【0015】**

好ましい実施態様においては、さらに、前記順序付き集合配列のサイズと、前記値リストのサイズとを比較するステップを備え、前記比較の結果、部分集合のサイズが、値リストのサイズより所定の割合で小さい場合に、前記 c) および d

）、または、前記 c）および e）のステップを実行する。

#### 【0016】

ここでは、第2のポインタ配列および第2の値リストを生成するための処理時間を考慮しても、処理時間の短縮化が認められる場合に、第2のポインタ配列および第2の値リストを生成し、これらを利用した処理を実行するように構成している。所定の割合は、たとえば、（部分集合のサイズ、つまり、レコード数）／（値リストのサイズ、つまり、レコード数）＝ $1/n$ （ $n$ は1より大きな数）にて定めることができる。この「 $n$ 」は、オペレータが所望の数を入力することにより定められても良いし、あらかじめ所定の数に定められていても良い。

#### 【0017】

また、本発明の目的は、上記データ処理方法により生成された第2のポインタ配列および／または第2の値リストを利用した検索方法であって、第2の値リスト中の要素のうち、検索対象となる項目値を指し示す要素を特定するステップと、前記第2の値リストと同じサイズのフラグ配列中、前記特定された要素の位置と対応する位置に、フラグがオンである状態を示す値を配置するステップと、前記順序付き集合中のレコード番号が指し示す、第2のポインタ配列の要素を特定するステップと、前記フラグ配列中、前記第2のポインタ配列の要素が指し示す位置のフラグの状態を参照するステップと、前記フラグの状態がオンである場合、前記レコード番号を、新たに設けられた出力用の順序付き集合配列に、順次配置するステップとを備えたことを特徴とする方法によっても達成される。

#### 【0018】

別の実施態様において、上記データ処理方法により生成された第2のポインタ配列および／または第2の値リストを利用した集計方法は、前記第2の値リストの要素に対応して、値のカテゴリを示す分類番号を配置する分類番号配列を生成するステップと、指定された分類番号に対応する、前記順序付き集合配列中のレコード番号を特定するステップと、前記特定されたレコード番号が指し示す、所定の値リストの項目値を用いた集計を実行するステップとを備えている。

#### 【0019】

さらに別の実施態様において、上記データ処理方法により生成された第2のポ

インタ配列および／または第2の値リストを利用したソート方法は、前記第2の値リストの値ごとに、前記第2のポインタ配列の指し示す要素の数である存在数を算出するステップと、前記存在数に基づき、前記第2の値リストの値に対応する、前記順序付き集合配列中のレコード番号を配置すべき先頭の位置を示す累計数配列を生成するステップと、前記累計数配列を参照して、前記値リスト中の項目値のソート順が反映されるように、前記順序付き集合配列のレコード番号を、出力用の配列中に配置するステップとを備えている。

#### 【0020】

また、別の実施態様において、上記データ処理方法により、生成された、複数の表形式データのそれぞれに関する情報ブロックにおける、第2のポインタ配列および／または第2の値リストを利用して、表形式データを結合する方法は、複数の表形式データのそれぞれにおいて、共有化すべき項目を見出すステップと、当該項目に関する情報ブロックの第2の値リスト中の項目値を等価にするステップと、前記項目値を等価にすることに応答して、それぞれの情報ブロックにおいて、第2のポインタ配列中の要素を、前記項目値の配置の変化に合致するように更新するステップとを備えている。

#### 【0021】

さらに、本発明の目的は、各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するために、コンピュータを動作させるデータ処理プログラムであって、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目

値番号配列中に配置するステップと、

d) 前記項目値番号配列中の値を参照して、位置の要素を格納した第2のポインタ配列および値の要素を格納した第2の値リストを生成するステップとを、前記コンピュータに実行させ、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定されるように、前記コンピュータを動作させることを特徴とするデータ処理プログラムによっても達成される。

#### 【0022】

好ましくは、前記第2のポインタ配列および第2の値リストを生成するステップにおいて、前記項目値番号配列中の要素をソートし、かつ、重複値が排除された状態となった配列を生成し、これを第2の値リストとするステップと、前記ソートおよび重複値の排除を反映するように、前記項目値番号配列中の要素を変換して、第2のポインタ配列を生成するステップとを前記コンピュータに実行させる。

#### 【0023】

別の実施態様において、各々が項目と当該項目に属する項目値とを含むレコードの配列として表される表形式データ中、処理対象とすべき部分集合を抽出するために、コンピュータを動作させるデータ処理プログラムは、

a) 前記表形式データを、各々が、特定の項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストと、一意的なレコード番号の順に、当該項目値番号を指示するためのポインタ値が格納されたポインタ配列とからなる情報ブロックに分割するように構成するステップと、

b) 前記レコードのうち、その一部が部分集合として選択されたことに応答して、当該部分集合のレコード番号を収容した順序付き集合配列を生成するステップと、

c) 前記順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値を、前記レコード番号の配置された位置に対応して、項目

値番号配列中に配置するステップと、

e) 前記項目値番号配列中を第2の値リストとして利用するとともに、もとの値リストを第2の値リストとして利用するステップとを、前記コンピュータに実行させ、

前記順序付き集合配列のレコード番号から、当該レコード番号が指し示す位置の第2のポインタ配列中の要素、および、第2のポインタ配列中の要素が指し示す位置の前記第2の値リスト中の要素を介して、値リスト中の値が特定されるように、前記コンピュータを動作させる。

#### 【0024】

好ましくは、さらに、前記順序付き集合配列のサイズと、前記値リストのサイズとを比較するステップを、前記コンピュータに実行させ、前記比較の結果、部分集合のサイズが、値リストのサイズより所定の割合で小さい場合に、前記c) およびd)、または、前記c) およびe) のステップを、前記コンピュータに実行させる。

#### 【0025】

なお、本明細書において、種々の配列中に収容される数値を、「要素」や「値」と称するが、これらは表現上の相違にすぎず、本質的な相違は無い。たとえば、「値リスト」に収容されるものは、「項目値」とも表現するが、これも値リスト中の「要素」であることは明らかである。

#### 【0026】

##### 【発明の実施の形態】

以下、添付図面を参照して、本発明の実施の形態につき説明を加える。図3は、本発明の実施の形態にかかる表形式データ中の処理対象となる部分集合のための変換配列を作成する処理、並びに、検索、集計、ソートおよびジョインの処理を実現するコンピュータシステムのハードウェア構成を示すブロックダイアグラムである。本実施の形態においては、検索、集計、ソートおよびジョインの処理に先立って、図2(a)～(c)に示すような情報ブロックに関して、処理対象となるレコードおよび値リストの要素のみを含む配列を作成する。この処理を「事前処理」と称する。次いで、事前処理により作成された配列を作業領域として



、検索、集計などの各種処理を実行する。

#### 【0027】

図3に示すように、このコンピュータシステム10は、通常のものと同様の構成であり、プログラムを実行することによりシステム全体および個々の構成部分を制御するCPU12、ワークデータなどを記憶するRAM(Random Access Memory)14、プログラム等を記憶するROM(Read Only Memory)16、ハードディスク等の固定記憶媒体18、CD-ROM19をアクセスするためのCD-ROMドライバ20、CD-ROMドライバ20や外部ネットワーク（図示せず）と接続された外部端子との間に設けられたインタフェース（I/F）22、キーボードやマウスからなる入力装置24、CRT表示装置26を備えている。CPU12、RAM14、ROM16、外部記憶媒体18、I/F22、入力装置24および表示装置26は、バス28を介して相互に接続されている。

#### 【0028】

本実施の形態にかかる、前処理のプログラム、検索、集計、ソート、ジョインなどのプログラムは、CD-ROM19に収容され、CD-ROMドライバ20に読取られても良いし、ROM16に予め記憶されていても良い。また、いったんCD-ROM19から読み出したものを、外部記憶媒体18の所定の領域に記憶しておいても良い。或いは、上記プログラムは、ネットワーク（図示せず）、外部端子およびI/F22を経て外部から供給されるものであっても良い。

#### 【0029】

また、本実施の形態においては、前処理、高速な検索、集計、ソートおよびジョイン処理を実現するために、図2（a）～（c）に示すデータ形式の情報ブロックを生成する必要がある。この情報ブロック生成プログラムも同様に、CD-ROM19に収容され、ROM16に記憶され、或いは、外部記憶媒体18に記憶されても良い。或いは、これらプログラムは、ネットワーク（図示せず）を介して、外部から供給されても良いことはいうまでもない。また、本実施の形態において、情報ブロック生成プログラムにて生成されたデータ（情報ブロック）は、RAM14に記憶され、或いは、外部記憶媒体18の所定の領域に記憶される。

**【0030】**

次に、コンピュータシステム10にて実行される処理の概略について説明を加える。図4に示すように、本実施の形態においてはレコード全体或いはその一部を処理対象とする検索、集計或いはソートの処理により、レコードがより限定的に選択され（ステップ401）、さらに、選択されたレコードの集合を処理対象として、検索、集計、ソート或いはジョインの処理が開始されるときに（ステップ402）、一定の条件の下で（ステップ403）、事前処理が実行される（ステップ404）。この事前処理を、場合により「スコープ」の設定処理と称し、事前処理が行われた後に行われる検索等の処理を、「スコープを用いた」検索等の処理と称する（ステップ405）。

**【0031】**

なお、選択されたレコードの集合を「順序付集合」と称する。この順序付集合の配列には、選択されたレコード番号が、所定の順序（たとえば昇順）で配置される。事前処理においては、後の検索等の処理に利用するための第2のポインタ配列（PV2）および第2の値リスト（VL2）が生成される。この生成処理については後述する。第2のポインタ配列は、順序付集合の要素数と同じ要素を含み、順序付集合の配列中の要素に対応する第2の値リスト中の要素を指し示すようになっている。また、第2の値リストには、値リスト（VL）の格納位置番号が収容される。第2の値リストの要素の数は、順序付集合の配列中の要素の数以下である。つまり、順序付集合の配列中の要素（選択されたレコード）に対応する項目値の全てが異なる場合のみ、これら要素の数が一致する。したがって、順序付集合の配列中の要素数は、第2の値リストの要素数のとり得る最大値となる。

**【0032】**

そこで、本実施の形態においては、本来の値リスト（VL）の要素数（サイズ）が、順序付集合の配列中の要素数（サイズ）より十分に小さい場合には（ステップ403参照）、本実施の形態における第2のポインタ配列および第2のVLを用いた処理、つまり、スコープを用いた検索等の処理が実行される（ステップ405）。たとえば、順序付集合のサイズが、VLのサイズの $1/n$ （ $n$ :設定

された1より大きい値) 以下であるとすれば良い。この「n」は、レコード数、プロセッサの処理能力などにしたがって設定することができる。

#### 【0033】

このような判断は、以下の理由により実行される。スコープを用いた処理においては、事前処理(ステップ404)が必要であり、その実行には時間を要する。このため、順序付集合のサイズ(実際には第2の値リストのサイズ)が、VLのサイズより十分に小さく、事前処理の実行時間が加えられた場合であっても、通常の処理(ステップ406)を実行するのに比較して、全体の処理時間を短縮できると考えられる場合のみ、つまり、事前処理およびスコープを用いた検索等の処理により、通常の処理と比較して、処理時間を短縮できる場合のみ、ステップ404およびステップ406の処理を実行することが望ましいからである。

したがって、上記「十分に小さい」とは、事前処理を経ても処理時間が短縮できる程度に小さいことを意味し、絶対的な数の大小を定義するものではない。

#### 【0034】

次に、本実施の形態にかかる事前処理につき説明を加える。図5は、本実施の形態にかかる事前処理を示すフローチャートである。図5に示すように、順序付き集合の配列中の位置(格納位置番号)を指定するパラメータを初期化した後(ステップ501)、当該パラメータの示す順序付き集合の配列中の要素が特定される(ステップ502)。初期的には、図6において、順序付き集合の配列601中、格納位置番号が「0」番の要素「0」が特定される。

#### 【0035】

次いで、上記順序付き集合の配列中の要素が指し示す、ポインタ配列中の要素を参照して、当該要素が指し示す位置の項目値の番号(項目値番号)が特定される(ステップ503)。上述した初期的状態の例では、順序付き集合の配列中の要素「0」が指し示すポインタ配列中の要素「2」が特定される。この要素は、値リスト中の関連する要素(項目値)の位置を示す番号を意味しているため、これを項目値番号と称している。

#### 【0036】

この項目値番号は、別途設けられた項目値番号配列中、パラメータが指し示す

位置に配置される（ステップ504）。上記初期的状態の例では、ポインタ配列212中の要素（項目値番号）「2」が、項目値番号配列611中、パラメータ「0」が指し示す位置に格納される。なお、項目値番号配列は、順序つき集合の配列と同じ数の要素を備える。

#### 【0037】

このような処理を繰り返すことにより（ステップ505、506参照）、順序つき集合中の要素（選択されたレコード番号）ごとに、対応する項目値番号を収容した項目値番号配列を作成することができる（図6の符号611参照）。

項目値番号配列の生成を完了すると、コンパイル処理が実行される（ステップ507）。本実施の形態においては、レコードの部分集合が処理対象である場合に、当該部分集合に関連する値リスト中の項目値のみを参照している。したがって、処理対象となる範囲を限定することから、本書面において「スコープ」とも称する。

#### 【0038】

スコープを実現するためには、部分中の要素が参照する値リストの部分集合と、レコードの部分集合中の要素（レコード番号）と、値リストの部分集合との変換配列とを要する。前者が、第2の値リスト（VL2）であり、後者が第2のポインタ配列（PV2）となる。そして、第2の値リストおよび第2のポインタ配列を生成する処理をコンパイル処理と称する。

図7は、コンパイル処理の一例を示すフローチャートである。この例にかかるコンパイル処理においては、まず、項目値番号配列の複製が作成される（ステップ701）。図6に示すような情報ブロック201およびレコードの部分集合である順序つき集合の配列601に関する例では、図8（a）に示すように、項目値番号配列（元配列）と同じ要素を有するような複製配列が生成される。

#### 【0039】

次いで、複製配列がソートされた後（ステップ702および図8（b）参照）、ソートされた複製配列中の重複値が排除され、配列がコンデンスされる（ステップ703および図8（c）参照）。このようにして、要素がソートされかつ重複値が排除された配列が、第2の値リスト（VL2）となる。さらに、項目値番

号配列（元配列）中の各要素の、第2の値リスト中の位置が特定され、特定された位置を示す値が、元配列の格納位置番号の要素として、新たに用意した配列に格納される（ステップ704および図8（d）参照）。ステップ704にて要素が格納された配列が、第2のポインタ配列（PV2）となる。

#### 【0040】

項目値番号配列は、値の成分と位置の成分とが混在すると考えることができる。そこで、ステップ701～703の処理において、値の成分のみを抽出し、抽出された値の成分のみを含む第2の値リスト（VL2）が生成されている。また、引き続きステップ704において位置の成分を抽出し、位置の成分のみを含む第2のポインタ配列（PV2）が生成されている。

このようにして、コンパイル処理により生成された第2の値リストおよび第2のポインタ配列は、情報ブロックの一部として、後述する検索、集計、ソート、ジョインなどの処理に利用される。

#### 【0041】

次に、上記第2の値リストおよび第2のポインタ配列を利用した（スコープを利用した）検索処理について説明する。図9は、本実施の形態にかかるスコープを利用した検索処理を示すフローチャートである。この処理においては、まず、第2の値リスト（VL2）と同じサイズのフラグ配列領域が確保される（ステップ901）。このフラグ配列においては、値リストの要素（項目値）のうち、検索対象となったものが格納される位置（格納位置）と同じ位置に、「1」が配置され、他の位置には「0」が配置される。したがって、次のステップにおいて、フラグ配列中に「1」或いは「0」の値が配置される。

#### 【0042】

図10は、選択されたレコード（順序付き集合）の項目「ファン」に関して、「Team」という項目値をもつレコードを検索する例を示している。ここでは、第2の値リスト（VL2）の要素が特定され、当該要素が指し示す値リスト（VL）中の要素（項目値）が検索対象であるか否かが判断される。項目値が検索対象であれば、フラグ配列中、第2の値リスト（VL2）の要素の格納位置に対応する位置に、「1」を配置する。図10の例では、項目値「Team

」が検索対象であるため、フラグ配列中、第2の値リスト (VL 2) 中、値リスト (VL) の項目値「Team」を指し示す要素の格納位置 (格納位置番号「1」) に、値「1」が配置される。

#### 【0043】

このようにして、フラグ配列に値が配置されると、順序付き集合の配列中の位置8格納位置番号)を示すパラメータを初期化した後 (ステップ903)、当該パラメータが指し示す、順序付き集合の配列中の要素が特定される (ステップ904)。図11の例では、初期的には、順序付き集合の配列中、格納位置番号が「0」番の要素「0」が特定される。次いで、当該要素が指し示す、第2のポインタ配列 (PV 2) 中の要素が特定され、さらに、PV 2中の要素が指し示すフラグ配列中の値が特定される (ステップ905)。上記初期的状態の例では、第2のポインタ配列 (PV 2) 中の要素「0」が指し示す位置のフラグ配列の値は「0」であることが特定される。

#### 【0044】

次いで、当該特定されたフラグ値が「1」であるか否かが判断される (ステップ906)。フラグ値が「1」であれば (ステップ906でイエス(Yes))、順序付き集合の配列の要素が、設けられた出力用配列中に配置される (ステップ907)。図11の例においては、順序付き集合の配列中、格納位置番号「1」の要素「2」について、PV 2およびフラグ配列をたどると、フラグ配列中のフラグ値が「1」となる。したがって、出力用配列中に、上記順序付き周到の配列の要素「2」が格納される。

このような処理が、順序付き集合の格納位置番号に対する全てのパラメータに関して実行される (ステップ908、909参照)。出力用配列に得られた要素が、それぞれ、検索により抽出されたレコード番号となる。

#### 【0045】

このように、本実施の形態によれば、レコードの部分集合である順序付き集合中の要素に対応付けられた、値リスト中の要素を示す情報 (値リストの格納位置番号) を、第2の値リスト (VL 2) に格納し、検索処理において、このサイズと同じサイズのフラグ配列を作成し、フラグ配列中の値を参照して、検索対象と

なったレコードであるか否かを判断している。フラグ配列のサイズは、第2の値リストのサイズと等しいため、全てのレコードに関連する値リストのサイズと比較して十分に小さい。したがって、フラグ配列の作成や値の参照に要する時間を短縮することができ、これにより、検索処理の処理時間を十分に短くすることが可能となる。

#### 【0046】

次に、スコープを利用した集計処理につき説明を加える。集計処理は、ある特定の1以上の項目値をもつレコード数を算出するものである。本実施の形態では、第2の値リストと同じサイズの存在数配列を設け、当該存在数配列に、第2の値リスト（VL2）中の項目値に関連するレコードの数を格納しておき、この存在数配列中の要素を取り出すことにより、即座に、集計を実現することができる。

#### 【0047】

図12は、「項目」ファンに関する情報ブロックにおける第2のポインタ配列（PV2）、第2の値リスト（VL2）および存在数配列を示す図である。PV2において、格納位置番号「0」および「2」の2個の要素が、VL2の格納位置番号「0」に一致する。したがって、存在数配列中、格納位置番号「0」の要素は、「2」となる。その一方、PV2において、格納位置番号「1」の1個（単一）の要素が、VL2の格納位置番号「1」に一致する。したがって、存在数配列中、格納位置番号「1」の要素は、「1」となる。なお、存在数配列の生成については、次に述べるソート処理において、詳細に説明する。

#### 【0048】

本実施の形態によれば、集計処理においても、利用する存在数配列のサイズを、本来の値リストよりも十分に小さい第2の値リストと同じサイズにすることができる。したがって、存在数配列の作成に要する時間を短縮でき、これにより全体の処理時間を短くすることが可能となる。

#### 【0049】

次に、スコープを用いたソート処理につき説明を加える。図13および図14は、スコープを用いたソート処理を示すフローチャートである。図13に示すよ

うに、まず、第2の値リスト(VL2)と同じサイズで、かつ、初期値「0」が配置された初期的な存在数配列が生成される(ステップ1301)。次いで、第2のポインタ配列(PV2)の位置(格納位置番号)を指定するためのパラメータを初期化した後(ステップ1302)、順序付き集合の配列中、パラメータが指し示す位置の要素が特定され、次いで、存在数配列中、当該要素が示す位置の値がインクリメントされる(ステップ1303)。ステップ1303は、最終のパラメータに達するまで繰り返される(ステップ1304、1305参照)。ステップ1305にてイエス(Yes)と判断された段階で、存在数配列が完成する。

#### 【0050】

図15および図16は、項目「ファン」に関する存在数配列および後述する累計数配列の生成過程の例を示す図である。図15(a)に示すように、初期的には、パラメータ「0」の位置のPV2の要素「0」が特定され、存在数配列中、当該PV2の要素が示す位置の値が「0」から「1」にインクリメントされる。次いで、図15(b)に示すように、パラメータ「1」の位置のPV2の要素「1」が特定され、存在数配列中、当該PV2の要素が示す位置の値が、「0」から「1」にインクリメントされる。さらに、図16(a)に示すように、パラメータ「2」の位置のPV2の要素「0」が特定され、存在数配列中、当該PV2の要素が示す位置の値が、「1」から「2」にインクリメントされる。

#### 【0051】

存在数配列が完成すると、存在数配列に基づき、その要素を累計化した累計数を格納した累計数配列が生成される(ステップ1306)。累計数配列は、ある格納位置番号より前の格納位置番号に関して、トータルで幾つ要素が存在したかを示す。図16(b)の例では、累計数配列1612において、格納位置番号「0」の要素は、それより小さな格納位置番号が存在しないため「0」となり、格納位置番号「1」の要素は、それより小さな格納位置番号「0」の存在数(存在崇拝列1611中の要素)と等しく、「2」となる。また、この累計数配列においては、さらなる要素(格納位置番号「2」の要素)として、格納位置番号「2」より小さな格納位置番号の存在数の総和「3」が配置されても良い(符号1612のカッコ参照)。



**【0052】**

このようにして累計数配列が生成されると、累計数配列を用いて、新たに確保した出力用配列中にレコード番号を配置する。図14に示すように、まず、順序付き集合の配列中の位置（格納位置番号）を指定するパラメータを初期化した後（ステップ1401）、パラメータが指し示す順序付き集合の配列中の要素が特定され、次いで、第2のポインタ配列（PV2）中、特定された要素が示す位置の要素が特定される（ステップ1402）。

**【0053】**

次いで、累計数配列中、第2のポインタ配列（PV2）中の特定された要素が示す位置の値（累計数）が特定される（ステップ1403）。さらに、パラメータが示す順序付き集合の配列中の要素が、別途設けた出力用配列中、ステップ1403にて特定された累計数が示す位置に配置される（ステップ1404）。この後、累計数配列において、ステップ1403にて特定された累計数がインクリメントされる（ステップ1405）。

**【0054】**

ステップ1403にて特定される累計数は、当該累計数を特定するもととなった順序付き集合の配列中の要素、つまり、レコード番号を、出力用配列に格納すべき位置を示す。したがって、あるレコード番号を配置した後、次に同じ累計数が特定され、出力用配列にレコード番号を格納するために、ステップ1405において、累計数をインクリメントしておく。また、順序付き集合の配列中の要素が格納されるため、出力用配列は、順序付き集合の配列と同じサイズとなる。

**【0055】**

このような処理を、最終パラメータに達するまで繰り返す（ステップ1406、1407参照）。このようにして得られた出力用配列において、キーとなった項目に関して、ソートされたレコード番号が示される。

**【0056】**

図17ないし図19は、項目「ファン」をキーとしたソート処理の過程における配列の状態を示す図である。図17においては、初期的に、パラメータ「0」が示す順序付き集合の配列の要素（レコード番号）「0」が特定されている。し

たがって、P V 2 を介して、P V 2 中の関連する要素が示す位置の累計数配列の値（累計数）「0」が特定される。したがって、出力用配列中、累計数が示す格納位置番号「0」の位置に、順序付き集合配列の要素「0」が格納され、その後、累計数が「0」から「1」にインクリメントされる。

#### 【0057】

図18においても、同様に、パラメータ「1」が指し示す順序付き集合の配列の値「2」が、出力用配列中、P V 2 を経て得られた累計数配列中の累計数「2」に対応する位置に格納され、その後、累計数配列中の累計数が「2」から「3」にインクリメントされる。さらに、図19においても同様に、パラメータ「2」が指し示す順序付き集合の配列の値「9」が、出力配列中、P V 2 を経て得られた累計数配列中の累計数「1」に対応する位置に格納され、その後、累計数配列中の累計数が「1」から「2」にインクリメントされる。

このように、本実施の形態によれば、ソートの処理においても、処理に際して利用される累計数配列を、第2の値リストと同じサイズにすることができ、これにより、処理時間の短縮を図ることができる。

#### 【0058】

次に、スコープを用いたジョイン処理につき説明を加える。ジョイン処理とは複数のテーブルから、所定の項目をキーとして、当該複数のテーブルを共通化して結合することをいう。たとえば、図20（a）に示す野球愛好会会員の元のテーブル（元テーブル）2001から、項目「地域」の値が「T o k y o」であるようなレコードを抽出したテーブル（野球愛好会会員テーブル）2011を作成し、かつ、図20（b）に示すように、野球の試合予定の元テーブル2011から、項目「野球場」の値が「T o k y o」であるようなレコードを抽出したテーブル（試合予定テーブル）2012を作成した場合を考える。ジョインの処理では、野球愛好会会員テーブル2011中の項目「地域」および試合予定テーブル2012の項目「野球場」をキーとして、図21に示すような、キー項目が結合されたテーブルが生成される。ここに、ソート順が反映される（維持される）テーブル2101をマスタ側のテーブル、他方のテーブル2111をスレーブ側のテーブルと称する。

**【0059】**

ジョイン処理に際して、マスタ側およびスレーブ側とも、キーとなる項目にて所定の値を有するレコードを抽出し、テーブルを生成する。したがって、レコードを抽出した際に、本来の値リストの要素数（サイズ）が順序付き集合の配列中の要素数（サイズ）より十分に小さい場合に、スコープを用いたジョイン処理が実行されることになる。

**【0060】**

本実施の形態によれば、たとえば、試合予定元テーブルは、図22に示すような構造を備えている。すなわち、「球団」、「試合日」および「野球場」というそれぞれの項目に関して、値リストへのポインタ配列（PV）および値リスト（VL）を含む情報ブロックが設けられている。また、上述したように項目「野球場」の値が「T o k y o」であるようなレコードは、レコード番号が「0」、「4」、「9」および「10」を有している。したがって、上記レコード番号を昇順にて並べた配列が、順序付き集合の配列となる。

**【0061】**

図23は、項目「球団」の情報ブロックに関する事前処理（図5参照）の過程を説明する図である。図5のステップ501～505の処理により、項目値番号配列が生成され、さらに、コンパイル処理（図5のステップ507および図7参照）により、第2の値リスト（VL2）および第2のポインタ配列（PV2）が生成される。

**【0062】**

図24は、第2の値リストおよび第2のポインタ配列を利用したジョイン処理、つまり、スコープを利用したジョイン処理を示すフローチャートである。ジョイン処理においては、まず、キーとなる項目の値リストが共通化される（ステップ2401）。図25は、値リスト共通化の処理をより詳細に示すフローチャートである。値リスト共有化処理においては、まず、マスタ側の値リスト中の値とスレーブ側の値リスト中の値とが比較され、何れの値が小さいか（たとえば、アルファベットでは、何れが前方に位置するか）が判断される（ステップ2501）。

## 【0063】

図26は、マスタ側のテーブルを構成する項目「ファン」の情報ブロックおよびスレーブ側のテーブルを構成する項目「球団」の情報ブロックのコンパイル後の初期状態を示す図である。図26に示すように、それぞれの情報ブロックについて、共有後の値リスト（NewVL）の値を指し示すための新たな第2の値リスト（新VL2）が設けられ、かつ、第2の値リスト中、処理対象となる要素の位置（つまり、値を配置すべき位置）を示すポインタ（PUTポインタ）が配置される。このポインタが指し示す位置を、「PUT位置」と称する。同様に、共有後の値リスト中に値を配置すべき位置を示すポインタ（PUTポインタ）も用意される。

## 【0064】

初期的には、マスタ側の第2の値リスト（VL2）の先頭の要素「2」が指し示すマスタ側値リスト（VL）の値「G-e-a-m」と、スレーブ側の第2の値リスト（VL2）の先頭の要素「2」が指し示すスレーブ側値リスト（VL）の値「D-e-a-m」とが比較され、アルファベット順で「D-e-a-m<G-e-a-m」と判断される（図27参照）。

## 【0065】

次いで、比較結果小さいと判断された値（図27においては、「D-e-a-m」）が、共有後の値リスト（newVL）中、「PUTポインタ」が指し示す位置（PUT位置）に格納される（ステップ2502）。次いで、ステップ2502にて共有後の値リストに格納された値を有していた新たな第2の値リスト（図27においては、スレーブ側の「新VL2」）の、「PUTポインタ」が指し示す位置（PUT位置）に、共有後の値リスト（newVL）の「PUT位置」（図27においては、「0」）が格納される（ステップ2503）。これにより、「PUT位置」が格納された新たな第2の値リスト（新VL2）が、共有後の値リスト（newVL）の対応する要素を指し示すことになる。

次いで、「PUT位置」が格納された第2の値リスト（図27では、スレーブ側VL2）および共有後の値リスト（newVL）の「PUT位置」がインクリメントされる（ステップ2504）。

## 【0066】

このような処理を、マスタ側の第2の値リスト (VL2) およびスレーブ側の第2の値リスト (VL2) の末尾まで繰り返す (ステップ2505およびステップ2506参照)。図27の例に引き続き、図28に示すように、マスタ側の第2の値リスト (VL2) の先頭の要素「2」が指し示す値リスト「VL」の要素「G-team」が取り出され、その一方、スレーブ側の第2の値リスト (VL2) の2番目の要素「3」が指し示す値リスト「VL」の要素「G-team」が取り出される。両者を比較すると「G-team=G-team」であるため、「G-team」を、共有後の値リスト (newVL) の「PUT位置」に格納するとともに、マスタ側、スレーブ側双方の新たな第2の値リスト (新VL2) の「PUT位置」に、共有後の値リスト (newVL) の「PUT位置」を格納する。その後、それぞれの配列 (新VL2、newVL) の「PUT位置」がインクリメントされる。

## 【0067】

これに引き続いて、図29に示すように、マスタ側の第2の値リスト (VL2) の2番目の要素「3」が指し示す値リスト「VL」の要素「T-team」が取り出され、その一方、スレーブ側の第2の値リスト (VL2) の3番目の要素「5」が指し示す値リスト「VL」の要素「T-team」が取り出され、両者が比較される。ここでは、「T-team=T-team」であるため、「T-team」を、共有後の値リスト (newVL) の「PUT位置」に格納するとともに、マスタ側、スレーブ側双方の新たな第2の値リスト (新VL2) の「PUT位置」に、共有後の値リスト (newVL) の「PUT位置」を格納する。その後、それぞれの配列 (新VL2、newVL) の「PUT位置」がインクリメントされる。このようにして、マスタ側およびスレーブ側の新たな第2の値リスト (新VL2) および共有後の値リスト (NewVL) が完成する。

## 【0068】

なお、実際には、新たな第2の値リスト (新VL2) の領域を別途作成せず、処理の過程において、第2の値リストVL2を上書きしても良い。また、説明の便宜のため、共有後の値リスト (newVL) の領域を確保して値を配置しているが、これは必須ではなく、上記共有後の値リストのサイズのみを把握し、メモリ等

に記憶しておけば足りる。

#### 【0069】

値リストの共有化が終了すると、スレーブ側の存在数配列、マスタ側の累計数配列、および、スレーブ側の累計数配列が順次生成される（ステップ2403）。この処理について、図31ないし図34を利用して、より詳細に説明する。まず、スレーブ側の第2のポインタ配列（PV2）から取り出された要素が指し示す、第2の値リスト（新VL2）中の値が特定される。次いで、存在数配列中、当該第2の値リスト（新VL2）の値が指し示す位置の値がインクリメントされる。

#### 【0070】

図31（a）においては、「PV2」の先頭の要素「2」が指し示す「新VL2」の値「2」が特定され、次いで、存在数配列中、格納位置番号が「2」の値が「0」から「1」にインクリメントされる。同様に、「PV2」の2番目の要素「1」に関して、存在数配列中、格納位置番号が「1」の値が「0」から「1」にインクリメントされ（図31（b）参照）、「PV2」の3番目の要素「0」に関して、存在数配列中、格納位置番号が「0」の値が「0」から「1」にインクリメントされ（図32（a）参照）、かつ、「PV2」の4番目の要素「1」に関して、存在数配列中、格納位置番号が「1」の値が「1」から「2」にインクリメントされる（図32（b）参照）。このようにして、スレーブ側の存在数配列の最終形を得ることができる（符号3211参照）。

#### 【0071】

次いで、上記スレーブ側の存在数配列を利用して、マスタ側の存在数配列、および、当該存在数配列中の要素（存在数）を累計数化した累計数配列が生成される。このマスタ側の存在数配列の要素（存在数）は、マスタ側の第2のポインタ配列にて特定される、値リスト中の値が、スレーブ配列では幾つ存在したかを示す。

#### 【0072】

図33に示すように、マスタ側の第2のポインタ配列（PV2）の先頭の要素「0」により特定される新たな第2の値リスト（新VL2）の値は「1」である

。これを用いて、スレーブ側の存在数配列を参照すると、上記値「1」が指し示す位置の要素は「2」となる。したがって、存在数配列の先頭の要素として「2」が配置される。また、マスタ側の第2のポインタ配列（PV2）の2番目の要素「1」により特定される新たな第2の値リスト（新VL2）の値は「2」である。これを用いて、スレーブ側の存在数配列を参照すると、上記値「2」が指し示す市の要素は「1」となる。したがって、存在数配列の2番目の要素として「1」が配置される。同様に、マスタ側の第2のポインタ配列（PV2）の3番目の要素「0」により特定される新たな第2の値リスト（新VL2）の値は「1」である。これを用いて、スレーブ側の存在数配列を参照すると、上記値「1」が指し示す位置の要素は「2」となる。したがって、存在数配列の3番目の要素として「2」が配置される。

#### 【0073】

次いで、上記マスタ側の存在数配列に基づき、累計数配列が生成される。このマスタ側の累計数配列を、場合によって、「射影配列①」とも称する。4番目の位置に配された値「5」は、ジョインされたテーブルにおける全レコード数を示す。また、図34に示すように、スレーブ側の存在数配列に基づき、累計数配列が生成される。スレーブ側の累計数配列を、場合によって「射影配列③」とも称する。さらに、マスタ側の順序付き集合を、場合によって、「射影配列②」とも称する。

#### 【0074】

このように、種々の存在数配列および累計数配列が生成された後、スレーブ側のレコード番号を配置した順序付き集合がソートされる（ステップ2403）。図35および図36を参照して、より具体的に順序付き集合のソートを説明する。スレーブ側の順序付き集合（つまり、スレーブ側のレコード番号の集合）から要素（つまりレコード番号）が取り出され、引き続いて、当該要素（図35（a）においては「0」）が指し示す第2のポインタ配列（PV2）の要素（図35（a）においては「2」）が特定される。さらに、PV2の要素が指し示す新たな第2の値リスト（新VL2）の値（図35（a）においては「2」）が特定される。次いで、累計数配列中、当該新VL2の値が示す位置の値（累計数）（図

35 (a) においては「3」) が取り出され、ソート後の順序付き集合配列中、前記累計数が示す位置に、順序付き集合の要素(レコード番号)が配置される。その後、前記累計数が(たとえば、図35 (a) においては、「3」から「4」に) インクリメントされる。

#### 【0075】

図35 (b) に示すように、順序付き集合配列の2番目の要素「4」についても、同様に、第2のポインタ配列(PV2)、新たな第2の値リスト(新VL2)を経て、累計数配列において、2番目(格納位置番号「1」)の要素(累計数)「1」が特定される。したがって、ソート後の順序付き集合配列において、累計数「1」が示す位置に、もとの順序付き集合配列の要素「4」が配置される。その後、累計数が「1」から「2」にインクリメントされる。

#### 【0076】

同様に、図36 (a) に示すように、順序付き集合配列の3番目の要素「9」について、第2のポインタ配列(PV2)、新たな第2の値リスト(新VL2)を経て、累計数配列において、1番目(格納位置番号「0」)の要素(累計数)「0」が特定される。したがって、ソート後の順序付き集合配列において、累計数「0」が示す位置に、もとの順序付き集合配列の要素「9」が配置され、その後、累計数が、「0」から「1」にインクリメントされる。さらに、図36 (b) に示すように、順序付き集合配列の4番目の要素「10」について、第2のポインタ配列(PV2)、新たな第2の値リスト(新VL2)を経て、累計数配列において、2番目(格納位置番号「1」)の要素(累計数)「2」が特定される。したがって、ソート後の順序付き集合配列において、累計数「2」が示す位置に、もとの順序付き集合配列の要素「10」が配置され、その後、累計数が、「2」から「3」にインクリメントされる。このようにして、スレーブ側のレコード番号のソートが完了する。

#### 【0077】

次いで、ジョインされたテーブルのレコード番号に対応する、マスタ側のテーブルおよびスレーブ側のテーブルのレコード番号が算出される。ジョインされたテーブルのレコード番号の総数(レコード数)は、マスタ側の累計数配列(射影



配列①)の累計数の末尾に配された値に一致する。したがって、図31ないし図36等を参照して説明した例では、レコード数は「5」となる。

#### 【0078】

図37および図38は、ジョインされたテーブルのレコード番号から、対応するマスタ側およびスレーブ側のレコード番号をそれぞれ算出する処理（レコード番号算出処理）を示すフローチャートである。また、図39は、図33ないし図36に示す例について、図37および図38に示す処理を施した得られたマスタ側のレコード番号配列およびスレーブ側のレコード番号配列を示す図である。

以下、図33ないし図36にて得られた射影配列を利用して、図39に示すレコード番号の配列を取得する場合を例に、レコード番号算出処理を詳細に説明する。

#### 【0079】

まず、ジョインされたテーブルのレコード番号が取り出される（ステップ3701）。このレコード番号を、レコード番号算出処理において「値A」と称する。初期的には、図40に示すように、ジョインテーブルのレコード番号配列から、先頭の要素「0」が取り出される。

次いで、マスタ側累計数配列（射影配列①）中、ステップ3701で取り出された要素（レコード番号）以下の最大値およびその格納位置が特定される（ステップ3702）。この最大値を、レコード番号算出処理では、「値B」と称する。図40の例では、射影配列①において、レコード番号「0」以下の最大値は「0」であるため、「値B」が「0」であり、また、その格納位置が「0」と特定される。

#### 【0080】

その後、マスタ側の順序付き集合配列（射影配列②）中、上記格納位置の要素（マスタ側のレコード番号）が特定され、この要素（レコード番号）が、マスタ側レコード番号配列中、ジョインテーブル中の格納位置に対応する位置に配置される（ステップ3703）。図40の例では、射影配列②において、格納位置が「0」つまり先頭の値「0」が取り出され、これが、図39に示すマスタ側レコード番号配列の先頭の要素として格納される（符号3901参照）。このように

して、ジョインテーブルのレコード番号に対応するマスタ側のレコード番号が算出される。

#### 【0081】

次いで、「値A」と「値B」の差分がとられる（ステップ3704）。レコード番号算出処理において、この差分値を「値C」と称する。図40の例においては、「値C=0」となる（符合4010参照）。

#### 【0082】

また、マスタ側第2の値リスト（PV2）中、ステップ3702で特定された格納位置の要素が特定され（ステップ3705）、さらに、VL2の要素が指し示す位置の新たな第2の値リスト（新VL2）中の値が特定される（ステップ3706）。図40の例においては、PV2中、格納位置「0」の要素「0」が特定され、次いで、新VL2中、格納位置が「0」の値「1」が取り出される。

さらに、スレーブ側累計数配列（射影配列③）中、新たな第2の値リスト（新VL2）の値が指し示す位置の要素が特定される（ステップ3801）。この要素を、レコード番号算出処理において「値D」と称する。図40の例では、射影配列③において、新VL2の値「1」が示す位置の要素（累計数）「1」が、値Dとして特定される。

#### 【0083】

この後、「値C」と「値D」とが加算される（ステップ3802）。レコード場番号算出処理において、上記加算値を「値E」と称する。図40の例では、「値E=0+1=1」となる。

次いで、スレーブ側順序付き集合配列（射影配列④）中、「値E」が示す位置の要素（スレーブ側のレコード番号）が特定され、この要素（レコード番号）が、スレーブ側のレコード番号配列中、ジョインテーブル中の格納位置に対応する位置に配置される（ステップ3803）。図40の例では、射影配列④の「1」に対応する位置の値「4」が取り出され、これが、図39に示すスレーブ側レコード番号配列の先頭の要素として格納される（ステップ3911参照）。

#### 【0084】

ステップ3701～ステップ3803の処理を、順次、ジョインテーブルのレ

コード番号のそれぞれについて施すことにより（ステップ 3804、3805 参照）、ジョインテーブルの各レコード番号に対応する、マスタ側およびスレーブ側のレコード番号が算出され、それぞれのレコード番号が、レコード番号配列の、対応する位置に配置される。

図 41 ないし図 44 は、それぞれ、ジョインテーブルのレコード番号配列中、2 番目（格納位置番号「1」）ないし 5 番目（格納位置番号「4」）のレコード番号から、マスタ側およびスレーブ側のレコード番号を算出する過程を説明する図である。

#### 【0085】

図 41 において、ジョインテーブルのレコード番号配列の要素（レコード番号）は「1」であり、射影配列①において、「1」以下の最大値「0」およびその格納位置「0」が特定される（ステップ 3701、3702）。射影配列②中、上記格納位置「0」に示される位置の要素「0」が、マスタ側のレコード番号となり、マスタ側レコード番号配列中の対応する位置に格納される（ステップ 3703 および符号 3902 参照）。また、「値  $C = 1 - 0 = 1$ 」が算出される（ステップ 3704）。

#### 【0086】

その一方、マスタ側 PV2 中、格納位置「0」の要素「0」、新 VL2 中、PV2 の要素「0」が指し示す位置の値「1」が特定され、さらに、射影配列③中、新 VL2 の値「1」にて示される要素（累計数）「1」が特定される（ステップ 3705、3706 および 3801）。次いで、「値  $E = 1 + 1 = 2$ 」が算出され（ステップ 3802）、射影配列②中、格納位置「2」にある要素（レコード番号）「10」が取り出され、スレーブ側レコード番号配列中の対応する位置に格納される（ステップ 3803 および符号 3912 参照）。

#### 【0087】

図 42 において、ジョインテーブルのレコード番号配列の要素（レコード番号）は「2」であり、射影配列①において、「2」以下の最大値「2」およびその格納位置「1」が特定される（ステップ 3701、3702）。射影配列②中、上記格納位置「1」に示される位置の要素「2」が、マスタ側のレコード番号と

なり、マスタ側レコード番号配列中の対応する位置に格納される（ステップ3703および符号3903参照）。また、「値 $C = 2 - 2 = 0$ 」が算出される（ステップ3704）。

#### 【0088】

その一方、マスタ側PV2中、格納位置「1」の要素「1」、新VL2中、PV2の要素「1」が指し示す位置の値「2」が特定され、さらに、射影配列③中、新VL2の値「2」にて示される要素（累計数）「3」が特定される（ステップ3705、3706および3801）。次いで、「値 $E = 0 + 3 = 3$ 」が算出され（ステップ3802）、射影配列②中、格納位置「3」にある要素（レコード番号）「0」が取り出され、スレーブ側レコード番号配列中の対応する位置に格納される（ステップ3803および符号3913参照）。

#### 【0089】

同様に、図43において、ジョインテーブルのレコード番号配列の要素（レコード番号）は「3」であり、射影配列①において、「3」以下の最大値「3」およびその格納位置「2」が特定される（ステップ3701、3702）。射影配列②中、上記格納位置「2」に示される位置の要素「9」が、マスタ側のレコード番号となり、マスタ側レコード番号配列中の対応する位置に格納される（ステップ3703および符号3904参照）。また、「値 $C = 3 - 3 = 0$ 」が算出される（ステップ3704）。

#### 【0090】

その一方、マスタ側PV2中、格納位置「2」の要素「0」、新VL2中、PV2の要素「0」が指し示す位置の値「1」が特定され、さらに、射影配列③中、新VL2の値「1」にて示される要素（累計数）「1」が特定される（ステップ3705、3706および3801）。次いで、「値 $E = 0 + 1 = 1$ 」が算出され（ステップ3802）、射影配列②中、格納位置「1」にある要素（レコード番号）「4」が取り出され、スレーブ側レコード番号配列中の対応する位置に格納される（ステップ3803および符号3914参照）。

#### 【0091】

さらに、図44において、ジョインテーブルのレコード番号配列の要素（レコ

ード番号)は「4」であり、射影配列①において、「4」以下の最大値「3」およびその格納位置「2」が特定される(ステップ3701、3702)。射影配列②中、上記格納位置「2」に示される位置の要素「9」が、マスタ側のレコード番号となり、マスタ側レコード番号配列中の対応する位置に格納される(ステップ3703および符号3905参照)。また、「値C = 4 - 3 = 1」が算出される(ステップ3704)。

#### 【0092】

その一方、マスタ側PV2中、格納位置「2」の要素「0」、新VL2中、PV2の要素「0」が指し示す位置の値「1」が特定され、さらに、射影配列③中、新VL2の値「1」にて示される要素(累計数)「1」が特定される(ステップ3705、3706および3801)。次いで、「値E = 1 + 1 = 2」が算出され(ステップ3802)、射影配列②中、格納位置「2」にある要素(レコード番号)「10」が取り出され、スレーブ側レコード番号配列中の対応する位置に格納される(ステップ3803および符号3915参照)。

#### 【0093】

図45は、上述した処理の結果得られたマスタ側テーブルのレコード番号配列およびスレーブ側テーブルのレコード番号配列、並びに、ジョインされたビューなどを示す図である。マスタ側テーブルのレコード番号配列3900(中のレコード番号)にしたがって、マスタ側の元のテーブル(野球愛好会会員元テーブル4501)からレコードを抽出することにより、元テーブルの順序が維持されたテーブル(野球愛好会会員テーブル4521)が得られる。その一方、スレーブ側テーブルのレコード番号配列3910(中のレコード番号)にしたがって、スレーブ側の元のテーブル(試合予定元テーブル45119からレコードを抽出することにより、マスタ側のレコードと対応付けられたテーブル(試合予定テーブル45319を得ることができる。

#### 【0094】

このように、本実施の形態によれば、ジョインの処理においても、それぞれの射影配列を、第2の値リストと同じにすることができる。また、共有化された値リスト(共有化された新たな値リスト)も、せいぜい、マスタ側の第2の値リス

トとスレーブ側の第2の値リストを加えたサイズに留まる。したがって、射影配列中の要素を特定し、或いは、第2の値リストの値を特定する際の処理時間を短縮することが可能となる。

本発明は、以上の実施の形態に限定されることなく、特許請求の範囲に記載された発明の範囲内で、種々の変更が可能であり、それらも本発明の範囲内に包含されるものであることは言うまでもない。

#### 【0095】

たとえば、前記実施の形態において、集計処理として、存在数配列を作成し、存在数配列中の要素を取り出す場合について説明しているが、これに限定されるものではない。一般に、集計は、根源的に「0以上の複数の次元」と「0または1の測度」とに分解される。たとえば、「性別×体重別の体重の平均値」を考えると、次元は「2次元」（「性別」および「体重別」）、測度は「体重」となる。測度が「0」である場合には、要素の個数が、集計にて得られる値（集計値）となる。この場合には、存在数配列中の要素が集計値となる。そのほか、測度として、合計、最大、最小、平均などをとることができる。

#### 【0096】

次元としては、種々の分類が可能である。たとえば、年齢において、0～9歳を10歳未満として一つのカテゴリとすることができる。ここでは、値リストと同じ要素数を有する分類番号配列を備え、分類番号配列において、値リストの値（年齢）「0」～「9」に対応する位置の要素として「0」を配置し、他の要素として「1」を配置すればよい。無論、根源的には、区別可能な一つの値ごとに、1つのカテゴリを割り振ることができる。すなわち、年齢において、分類番号配列において、たとえば昇順に、「0」、「1」、「2」、・・・という値を配置することが可能である。そして、集計においては、ある分類番号に関連するようなレコード番号の集合を取得し、当該レコード番号から特定される値に対して、所定の演算を施せばよい。これにより、合計、最大、最小など種々の集計値を得ることが可能となる。本実施の形態においても、第2の値リストと同じサイズの分類番号配列を備え、そこに、カテゴリに対応する値を割り振れば、種々の集計値を取得することが可能である。

## 【0097】

また、前記実施の形態においては、第2のポインタ配列および第2の値リストを生成することで、検索、集計、ソート、ジョインなどに利用する配列のサイズを圧縮し、これにより、必要とするメモリ領域を小さくし、処理時間の短縮を図っている。しかしながら、これに限定するものではなく、第2のポインタ配列および第2の値リストを生成する中間段階に生成した項目値番号配列を利用しても良い。この場合には、第2のポインタ配列（PV2）の代わりに、項目値番号配列を利用し、第2の値リスト（VL2）の代わりに、もとの値リスト（VL）を利用すれば良い。つまり、上記実施の形態において、たとえば、値リスト中の項目値を特定するために、

## 【0098】

順序付き集合配列中の要素、第2のポインタ配列中の要素、第2の値リスト中の要素、値リスト中の要素（項目値）の順で、要素が特定されたのに対して、上記変形例では、順序付き集合配列中の要素、項目値番号配列中の要素、値リスト中の要素（項目値）の順で、要素が特定されれば良い。

また、検索、集計、ソート、ジョインなどの処理においても、値を参照する配列として、第2のポインタ配列の代わりに項目値番号配列を利用し、かつ、第2の値リストの代わりにもとの値リストを利用すれば良い。

## 【0099】

さらに、前記実施の形態においては、一般のコンピュータシステム10内に、所定のプログラムを読み込み、当該プログラムを実行することにより、複数の表形式データのジョインおよびジョインされた表形式データに関する処理を実現しているが、本発明はこれに限定されるものではなく、パーソナルコンピュータ等のような一般のコンピュータシステムに、データベース処理専用のボードコンピュータを接続し、当該ボードコンピュータが上記処理を実行できるように構成しても良いことは言うまでもない。したがって、本明細書において、手段とは必ずしも物理的手段を意味するものではなく、各手段の機能が、ソフトウェアおよび／またはハードウェアによって実現される場合を包含する。さらに、一つの手段の機能が、二つ以上の物理的手段により実現されても、若しくは、二つ以上の手

段の機能が、一つの物理的手段により実現されてもよい。

【0100】

【発明の効果】

本発明によれば、上記配列を部分集合の大きさ、これに伴って限定される値リストの部分集合の大きさに合致させて、部分集合の大きさに相当する処理時間の短縮を実現する手法を提供することが可能となる。

【図面の簡単な説明】

【図1】 図1は、表形式データの例、および、その部分集合の例を示す図である。

【図2】 図2は、図1の表形式データに基づく情報ブロックの例を示す図である。

【図3】 図3は、本発明の実施の形態にかかる処理を実現するためのプログラムを実行するコンピュータシステムのハードウェア構成を示す図である。

【図4】 図4は、本実施の形態において、コンピュータシステムにて実行される処理の概略を示すフローチャートである。

【図5】 図5は、本実施の形態にかかる事前処理をより詳細に示すフローチャートである。

【図6】 図6は、本実施の形態にかかる項目値番号配列の生成を説明するための図である。

【図7】 図7は、本実施の形態にかかるコンパイル処理の例を示すフローチャートである。

【図8】 図8は、本実施の形態にかかるコンパイル処理を説明するための図である。

【図9】 図9は、本実施の形態にかかる検索処理を示すフローチャートである。

【図10】 図10は、本実施の形態にかかる検索処理を説明するための図である。

【図11】 図11は、本実施の形態にかかる検索処理を説明するための図である。



【図 1 2】 図 1 2 は、本実施の形態にかかる検索処理を説明するための図である。

【図 1 3】 図 1 3 は、本実施の形態にかかるソート処理を示すフローチャートである。

【図 1 4】 図 1 4 は、本実施の形態にかかるソート処理を示すフローチャートである。

【図 1 5】 図 1 5 は、本実施の形態にかかるソート処理を説明するための図である。

【図 1 6】 図 1 6 は、本実施の形態にかかるソート処理を説明するための図である。

【図 1 7】 図 1 7 は、本実施の形態にかかるソート処理を説明するための図である。

【図 1 8】 図 1 8 は、本実施の形態にかかるソート処理を説明するための図である。

【図 1 9】 図 1 9 は、本実施の形態にかかるソート処理を説明するための図である。

【図 2 0】 図 2 0 は、本実施の形態にかかるジョインの概略を説明するための図である。

【図 2 1】 図 2 1 は、本実施の形態においてジョインされた表（ビュー）の例を示す図である。

【図 2 2】 図 2 2 は、本実施の形態において、ジョインされる表形式データに基づく情報ブロックを示す図である。

【図 2 3】 図 2 3 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 2 4】 図 2 4 は、本実施の形態にかかるジョイン処理の概略を説明するフローチャートである。

【図 2 5】 図 2 5 は、本実施の形態にかかる値リスト共有化処理を示すフローチャートである。

【図 2 6】 図 2 6 は、本実施の形態にかかるジョイン処理を説明するための

図である。

【図 27】 図 27 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 28】 図 28 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 29】 図 29 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 30】 図 30 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 31】 図 31 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 32】 図 32 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 33】 図 33 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 34】 図 34 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 35】 図 35 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 36】 図 36 は、本実施の形態にかかるジョイン処理を説明するための図である。

【図 37】 図 37 は、本実施の形態にかかるジョイン処理のうち、レコード番号算出処理を示すフローチャートである。

【図 38】 図 38 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 39】 図 39 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 40】 図 40 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 4 1】 図 4 1 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 4 2】 図 4 2 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 4 3】 図 4 3 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 4 4】 図 4 4 は、本実施の形態にかかるレコード算出処理を説明するための図である。

【図 4 5】 図 4 5 は、本実施の形態にかかるジョイン処理により最終的に得られる表（ビュー）を示す図である。

【符号の説明】

10	コンピュータシステム
12	CPU
14	RAM
16	ROM
18	固定記憶装置
20	CD-ROMドライバ
22	I/F
24	入力装置
26	表示装置

【書類名】 図面

【図 1】

図1

(a)

会員名	地域	ファン
0 Greene	Tokyo	G-Team
1 Bernstein	S-Chiba	G-Team
2 Carter	Tokyo	T-Team
3 Wilson	S-Chiba	B-Team
4 Bush	Kanagawa	B-Team
5 Eliot	Kyoto	T-Team
6 Hilton	Kanagawa	B-Team
7 Jackson	E-Osaka	T-Team
8 Smith	Aichi	D-Team
9 Fitzgerald	Tokyo	G-Team

100

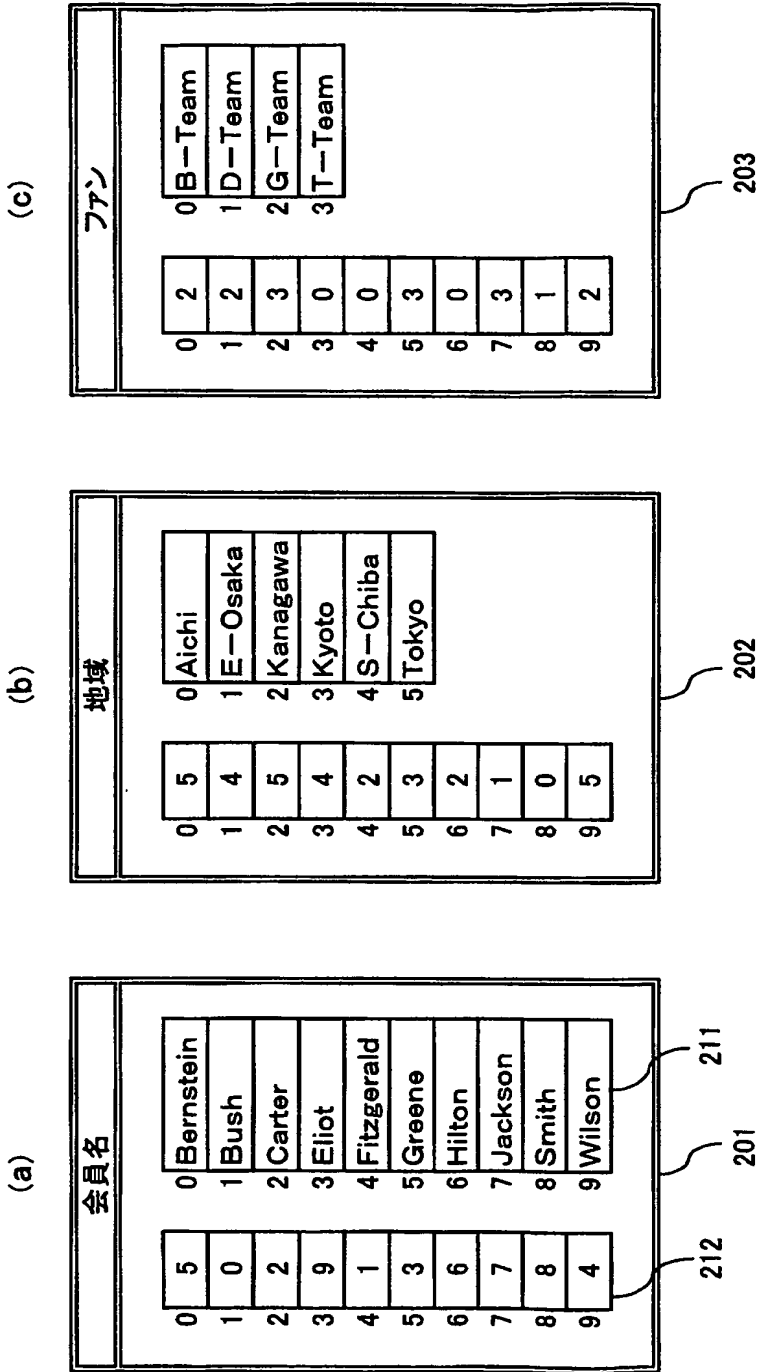
101

(b)

会員名	地域	ファン
0 Greene	Tokyo	G-Team
2 Carter	Tokyo	T-Team
9 Wilson	Tokyo	G-Team

0 0  
1 2  
2 9

【図 2】



【図 3】

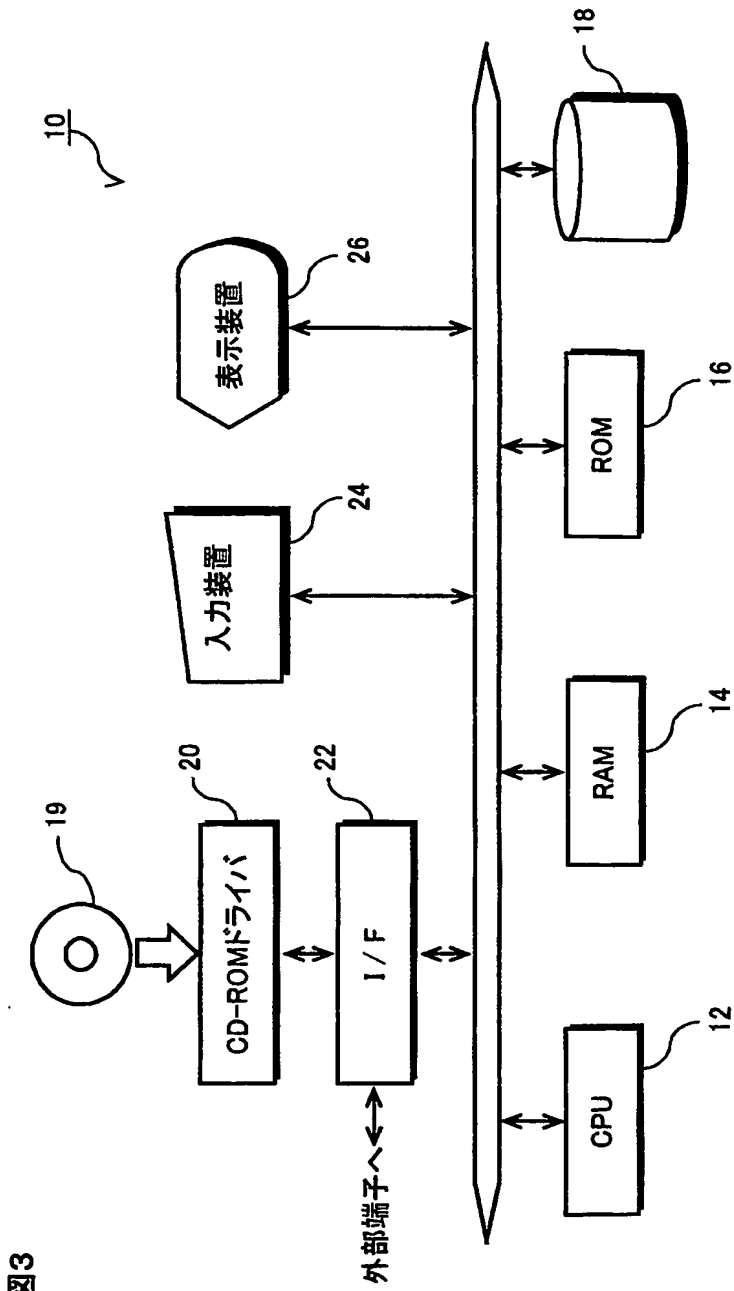
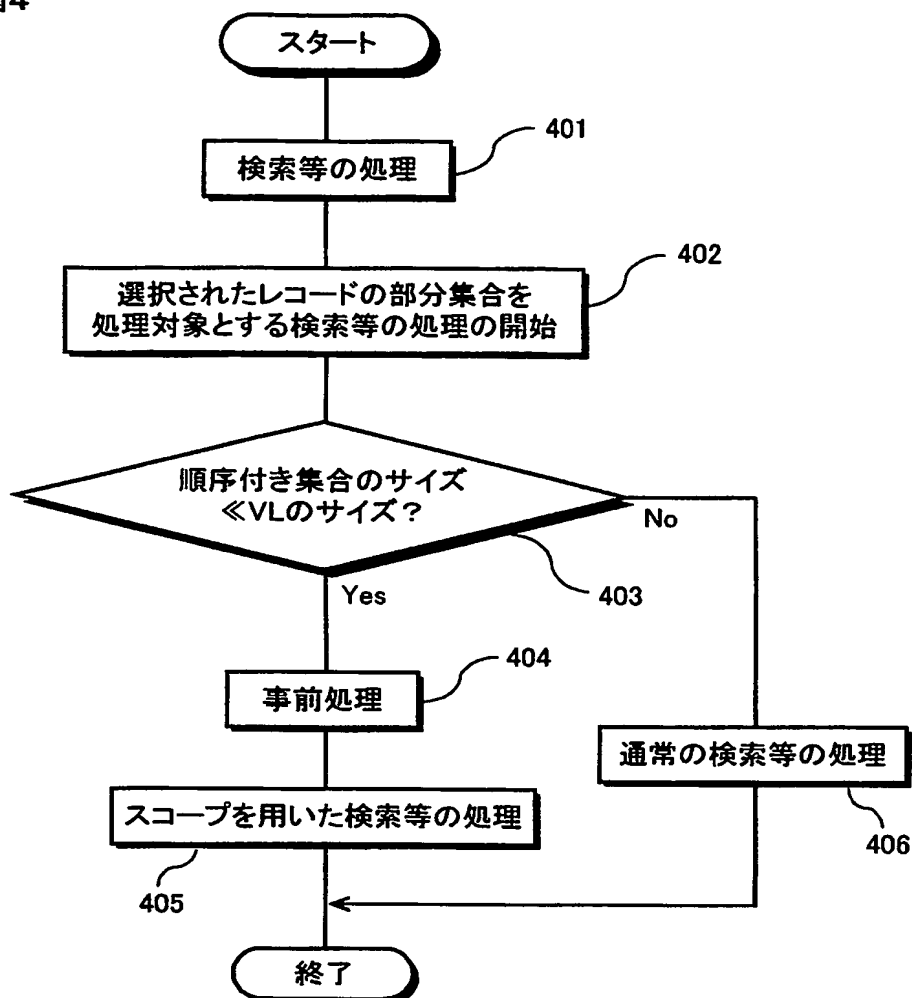


図 3

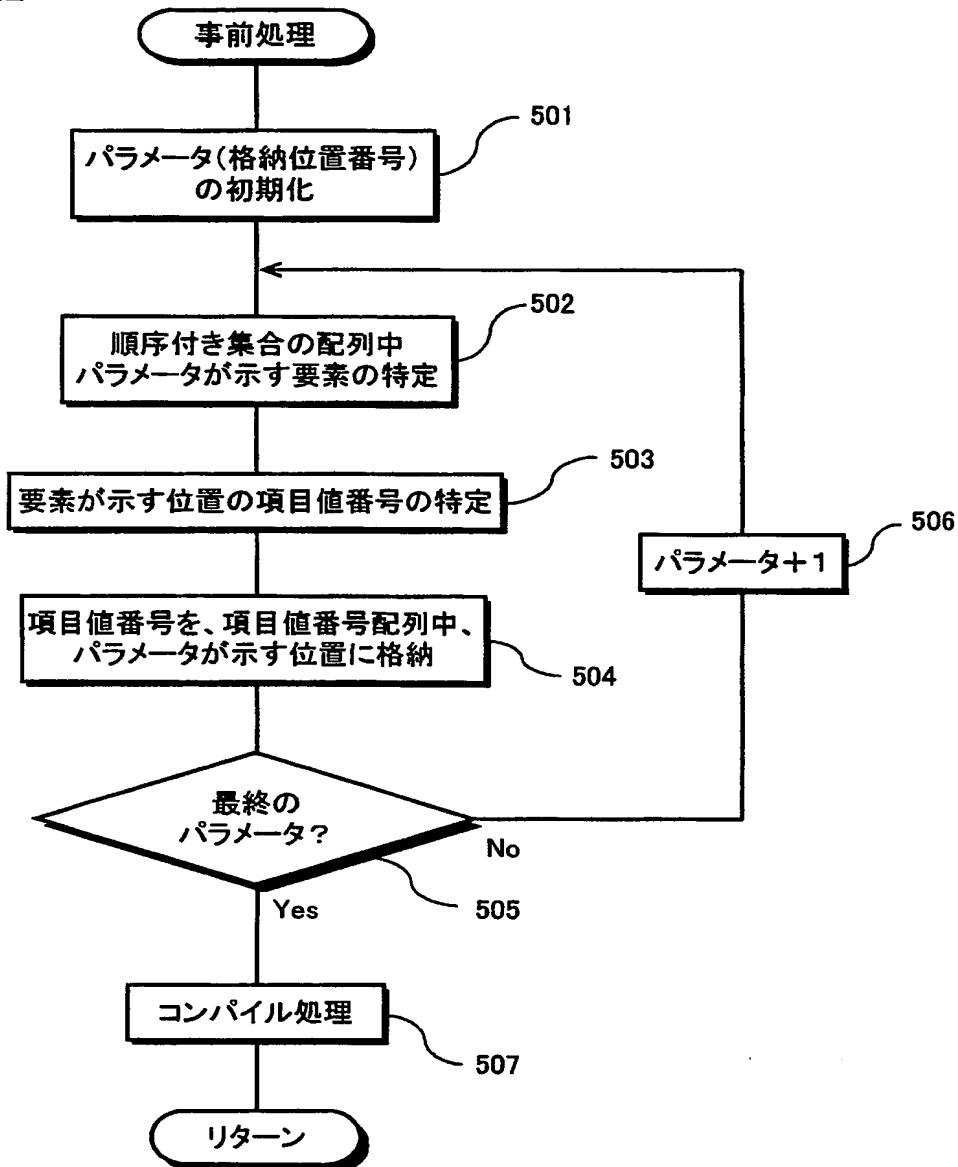
【図 4】

図 4



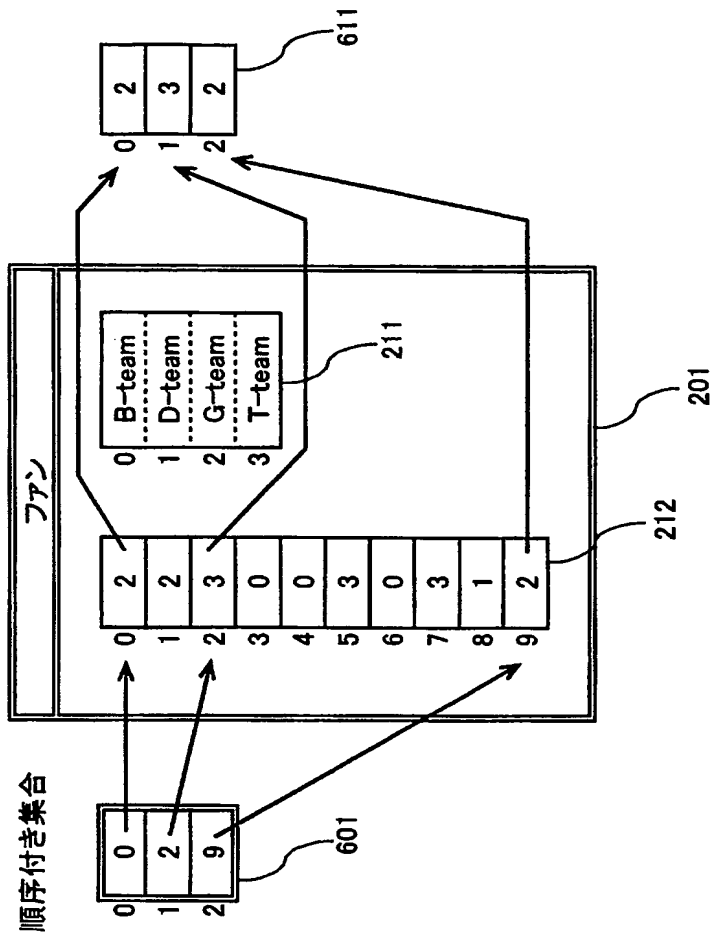
【図 5】

図5



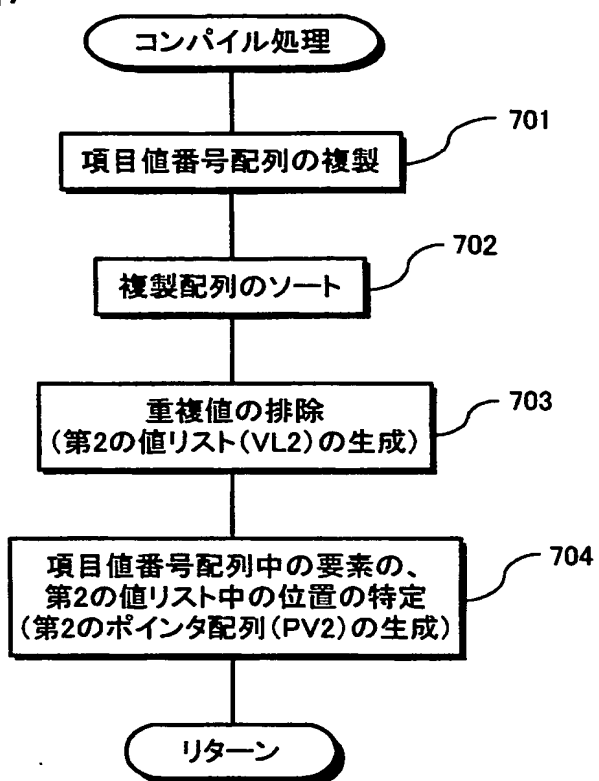


【図 6】



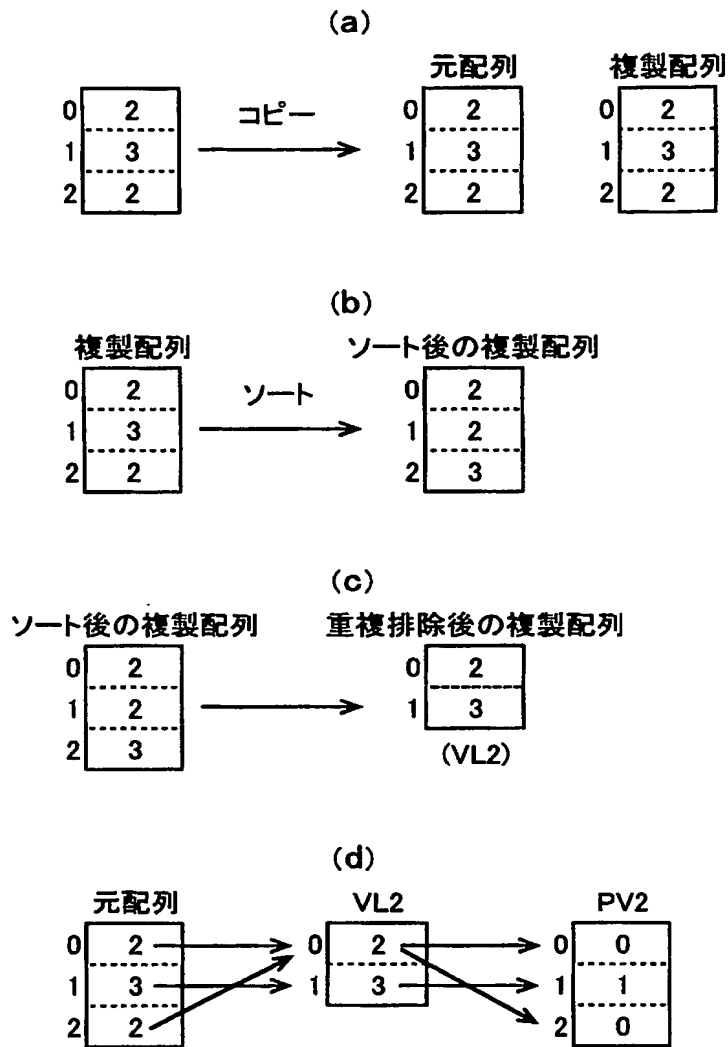
【図 7】

図 7



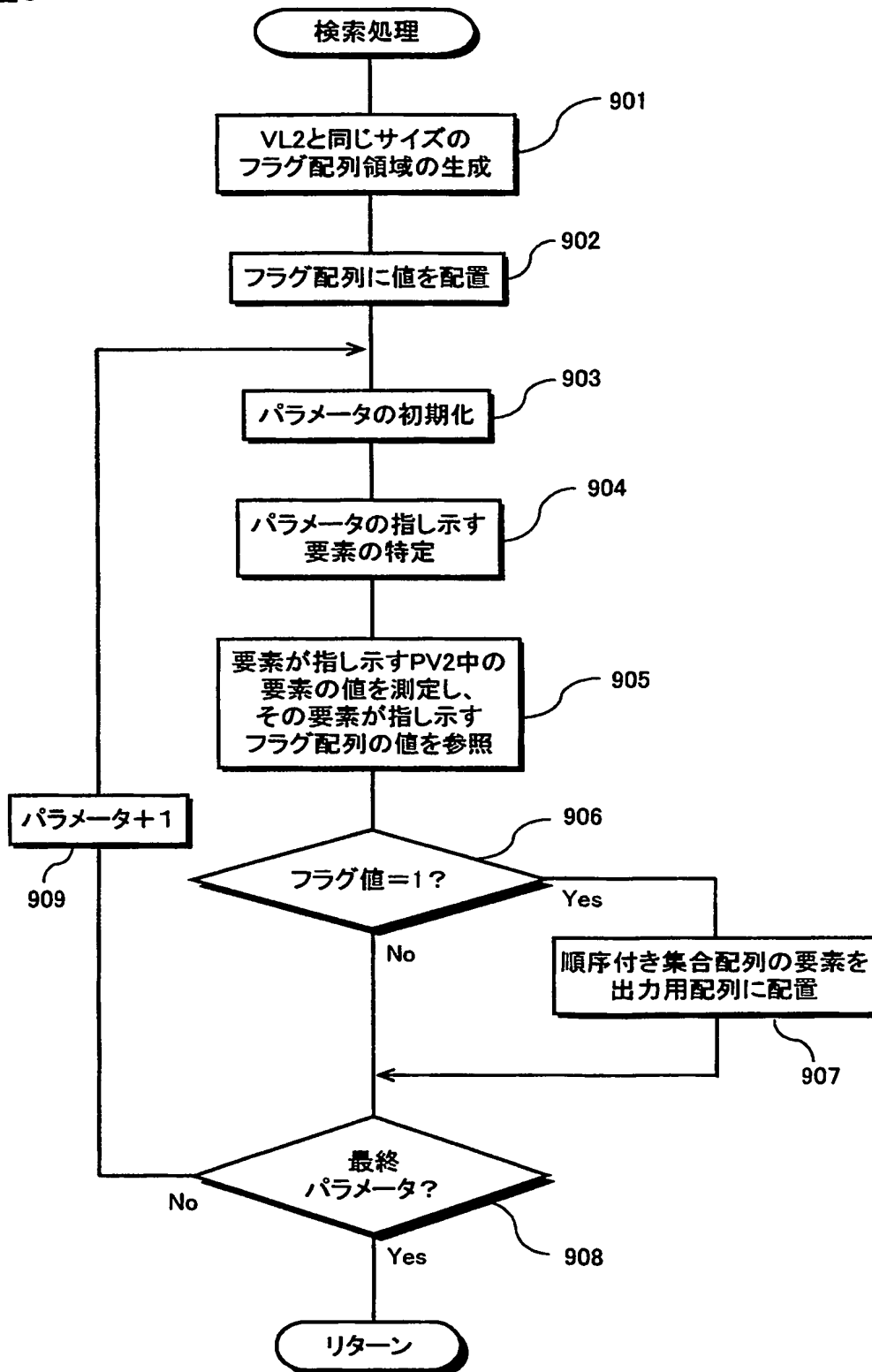
【図 8】

図8

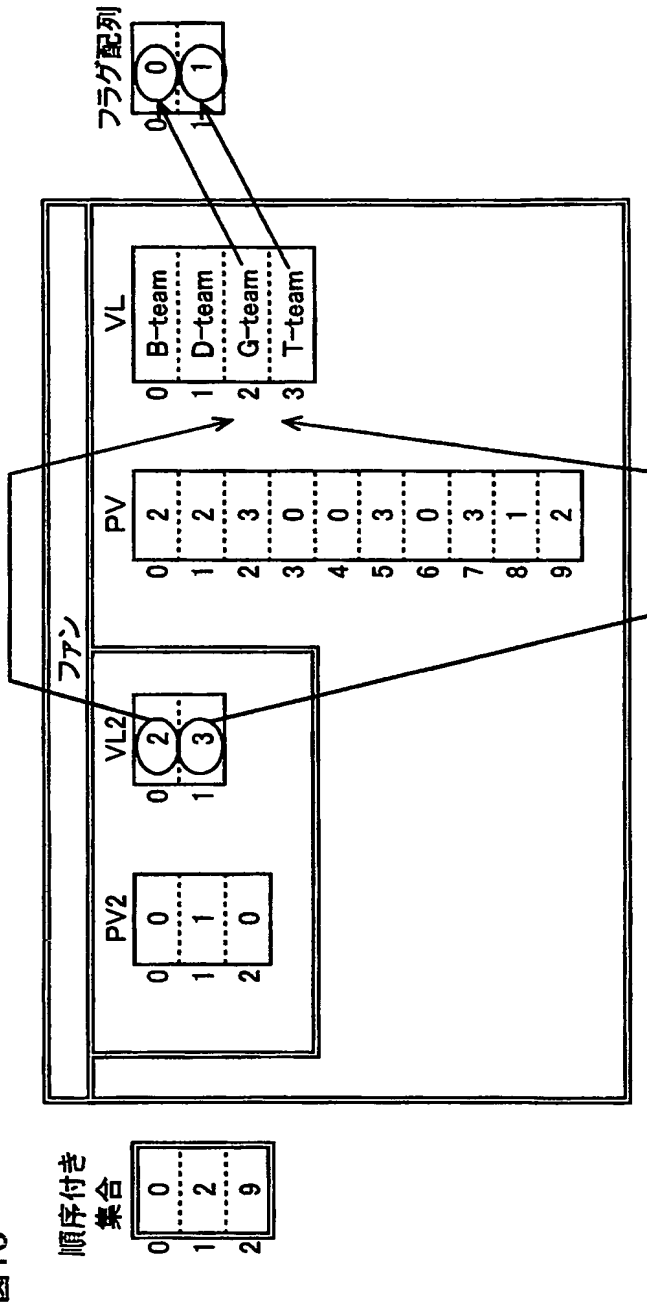


【図 9】

図 9



【図 10】



【図 11】

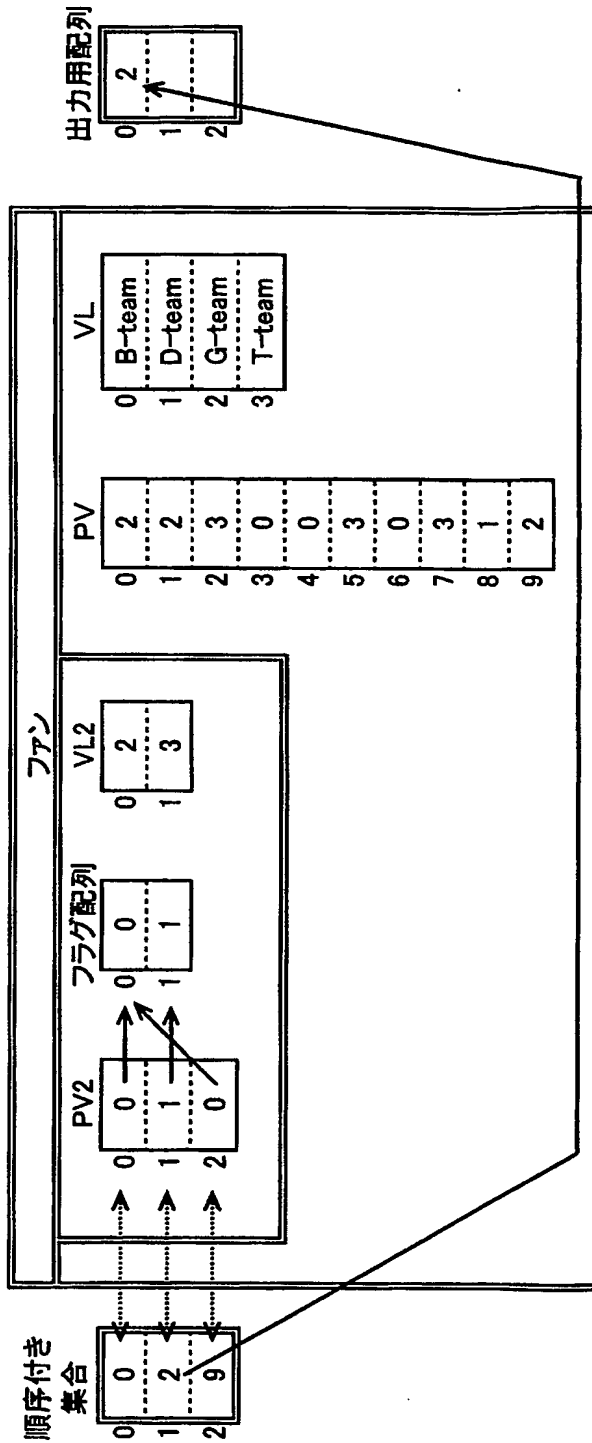


図 11

【図 12】

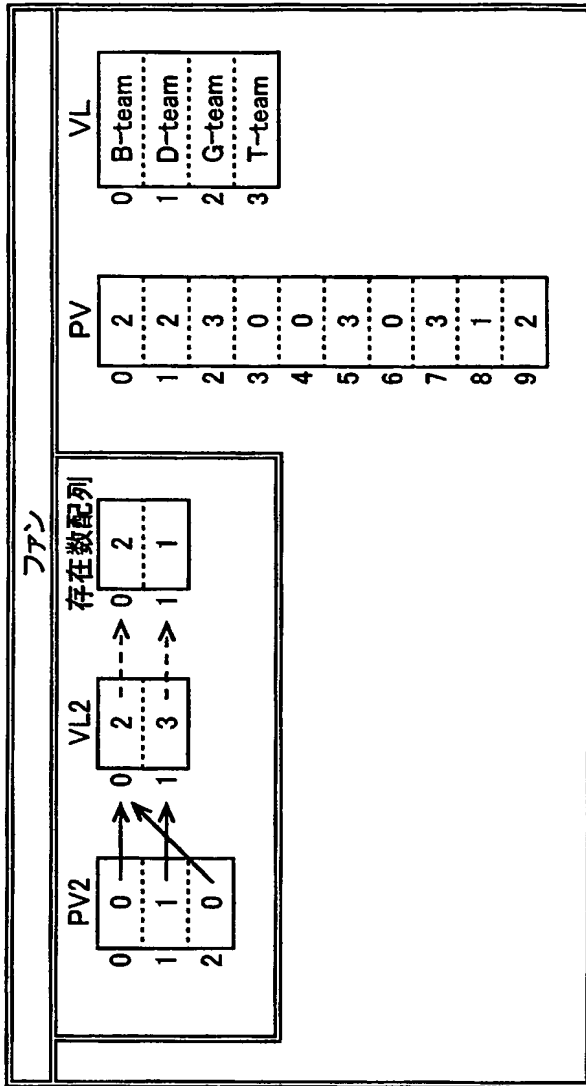


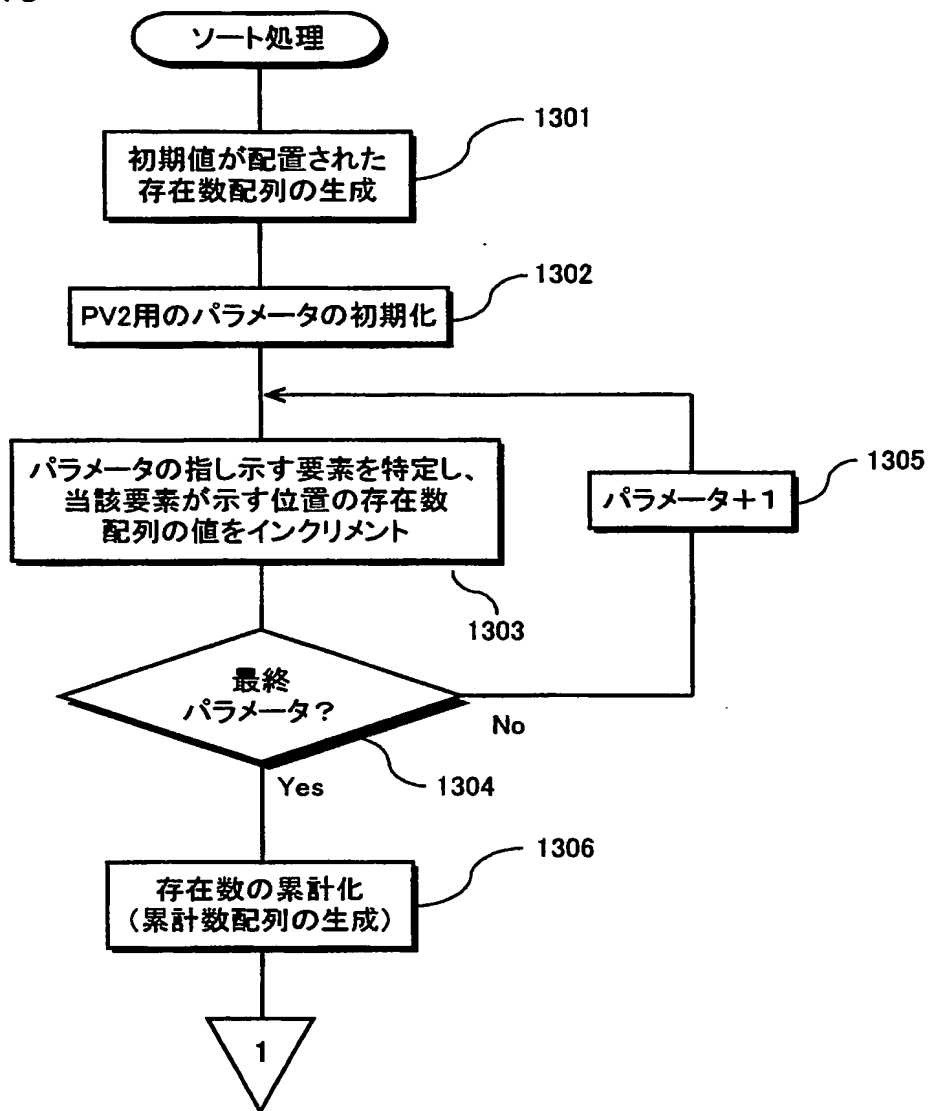
図12

順序付き  
集合

0	0
1	2
2	9

【図 13】

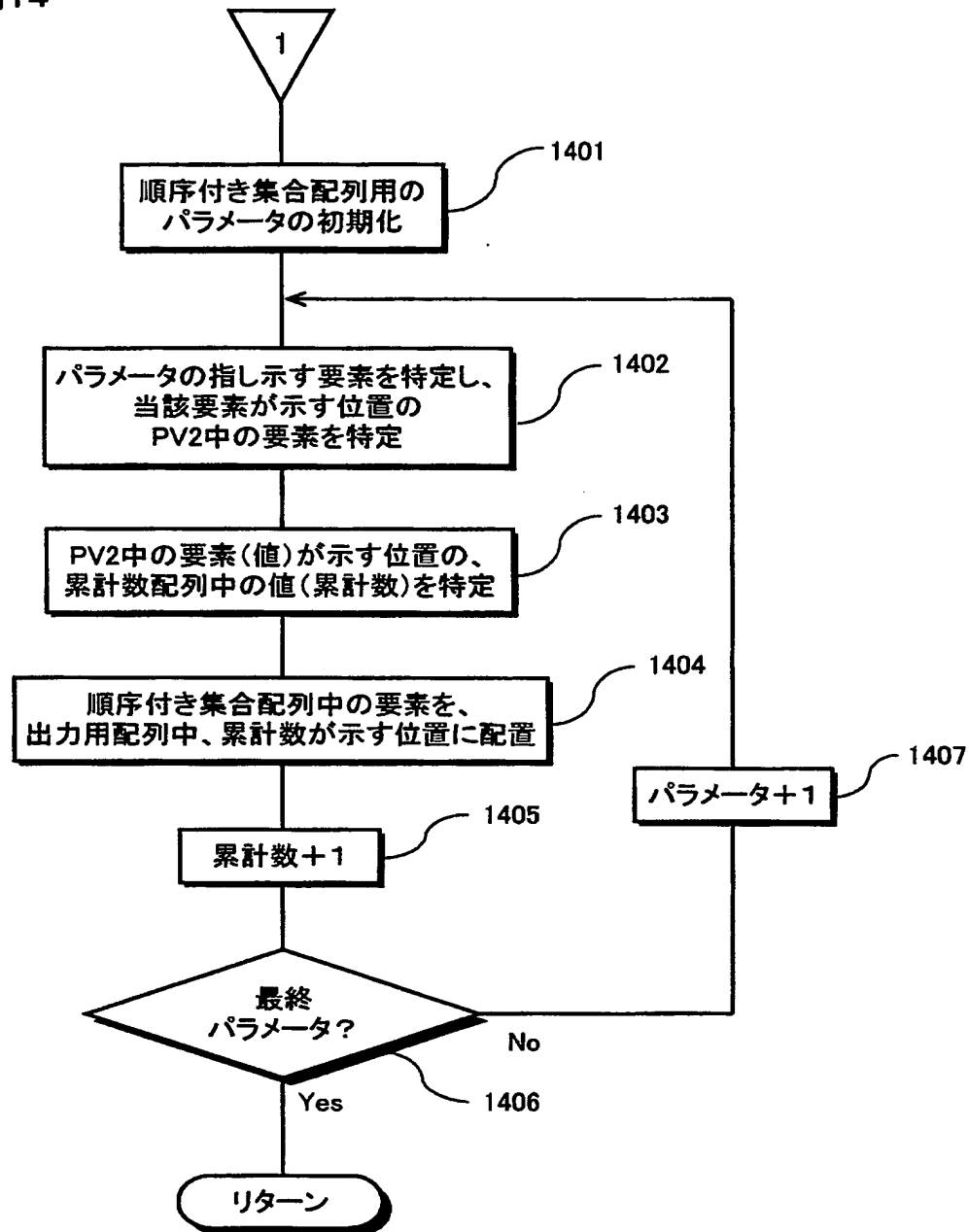
図13





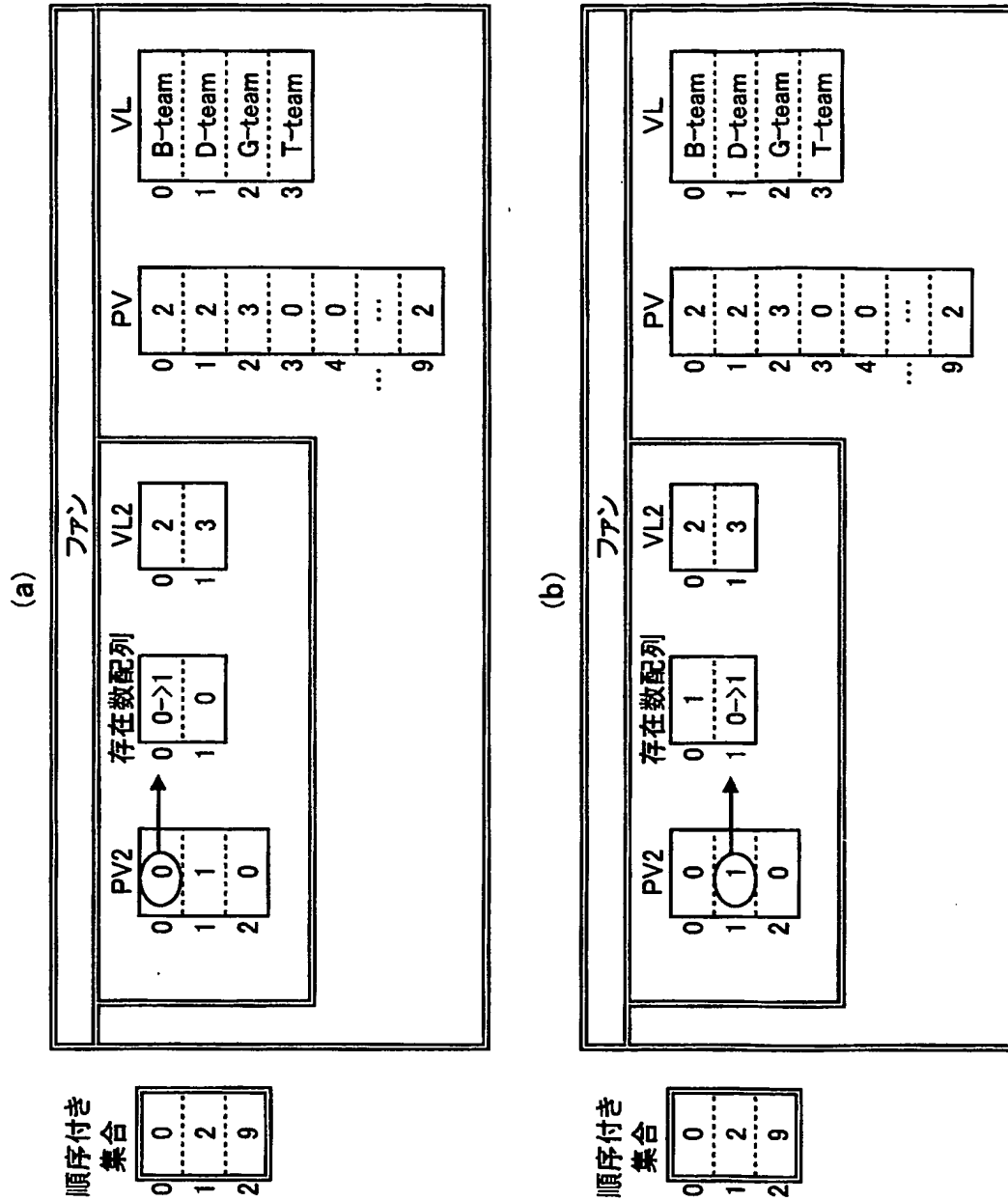
【図 14】

図14

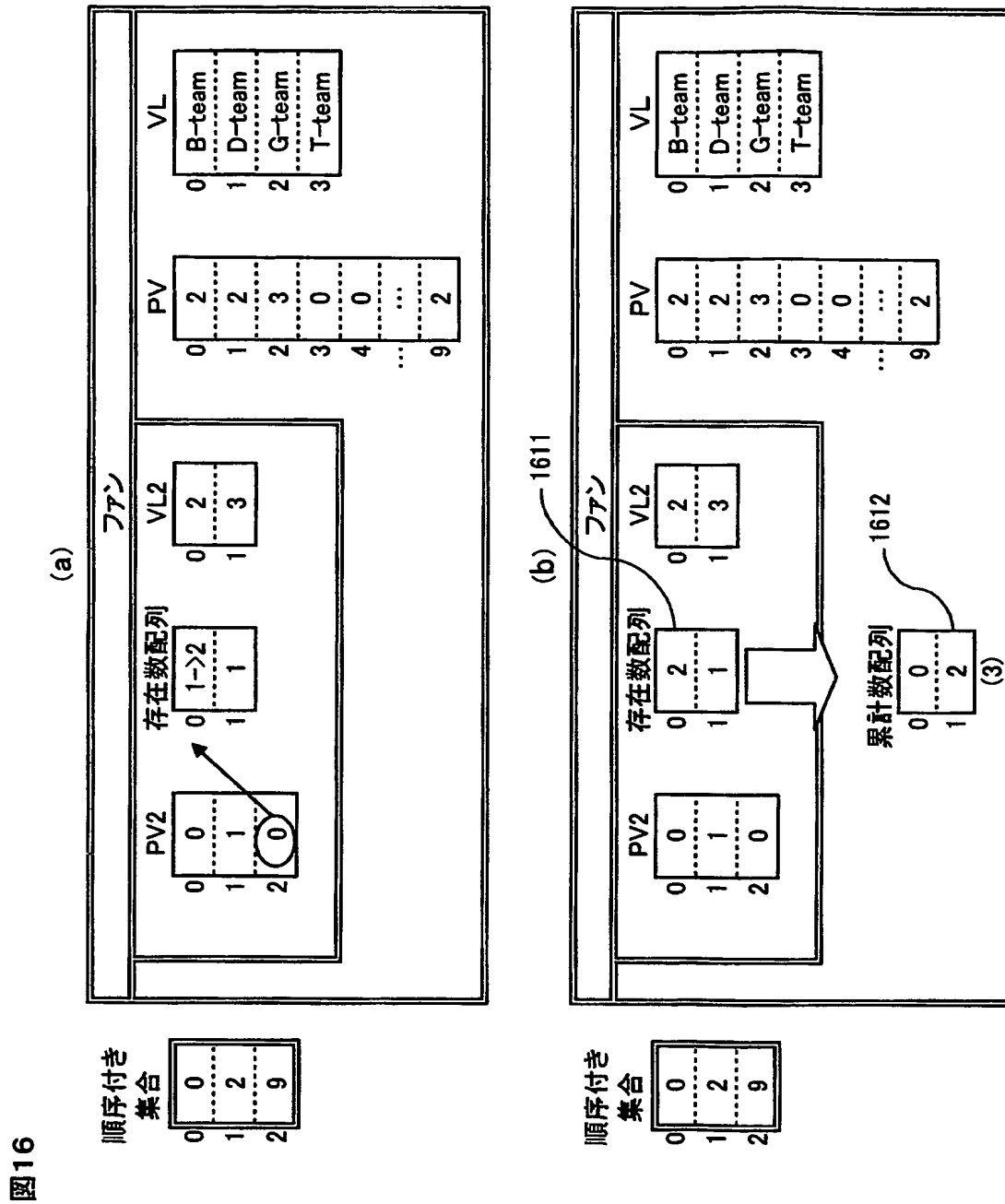


【図15】

図15



【図 16】



【図 17】

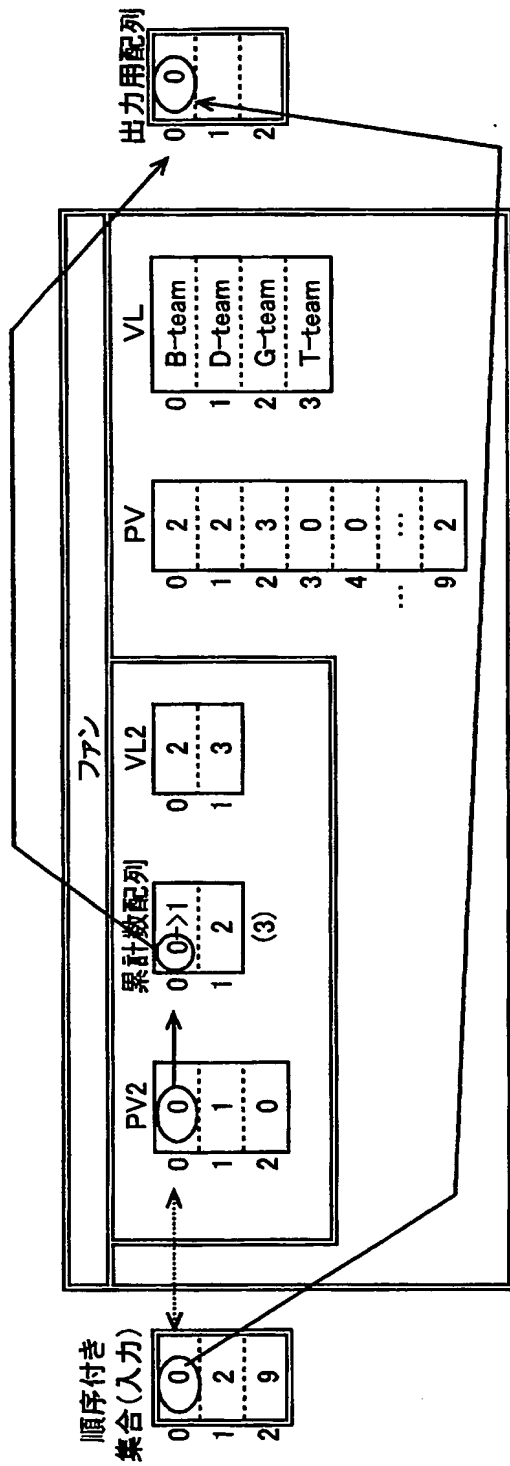
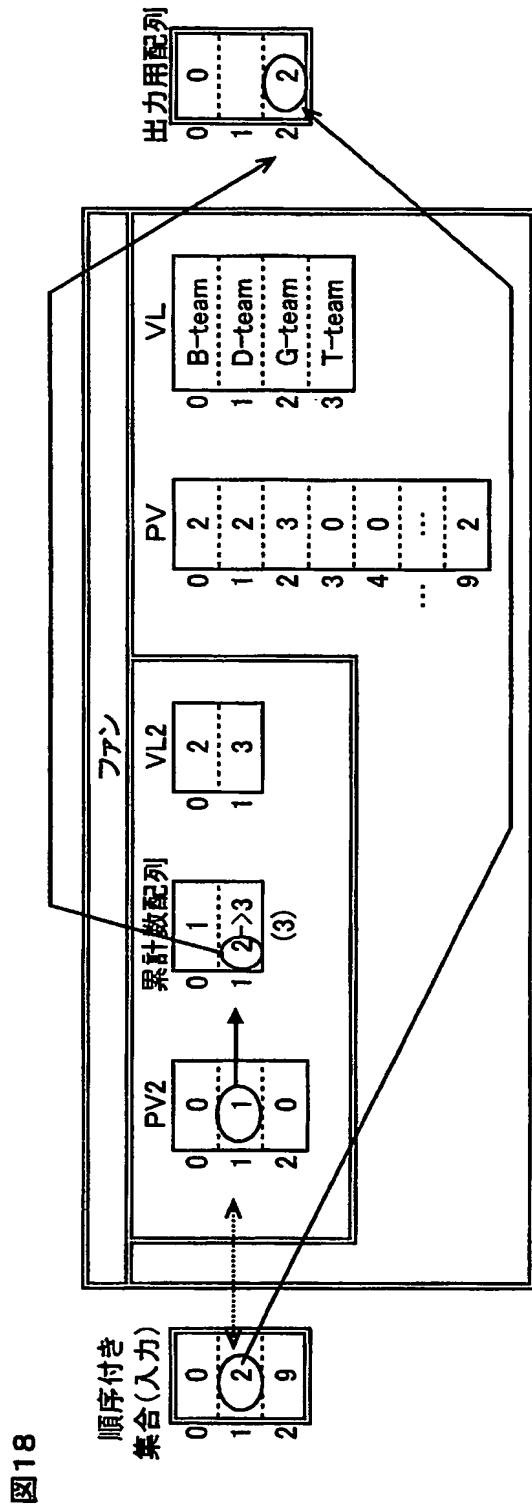
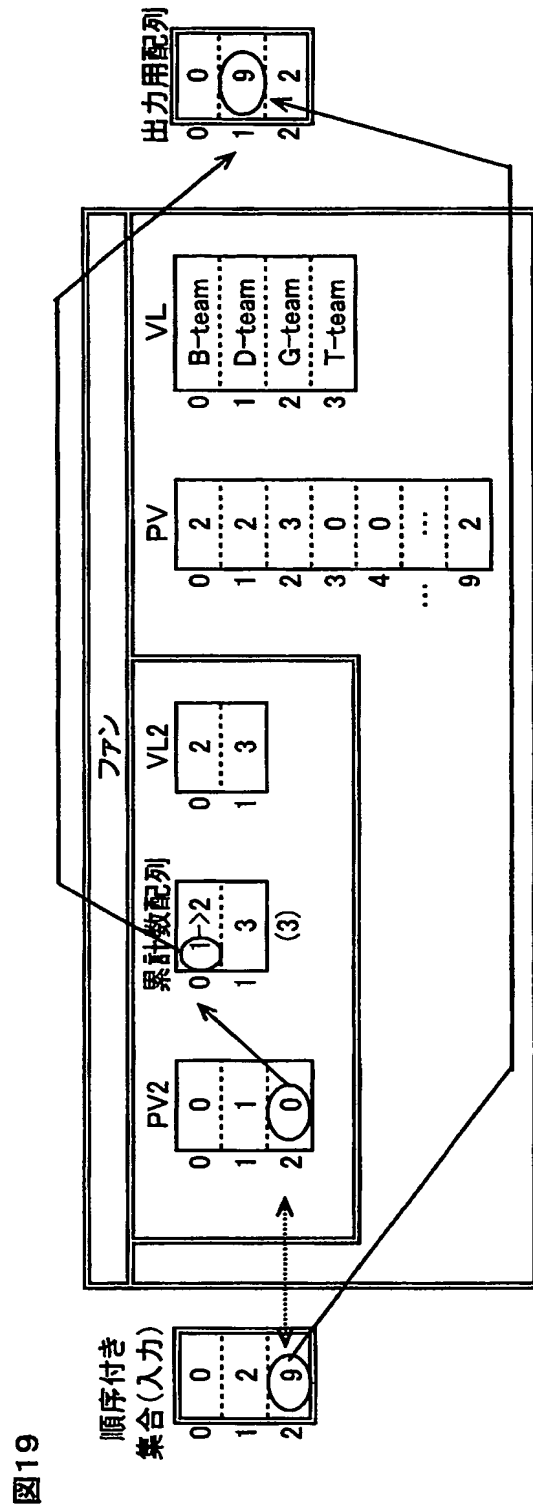


図17

【図 18】



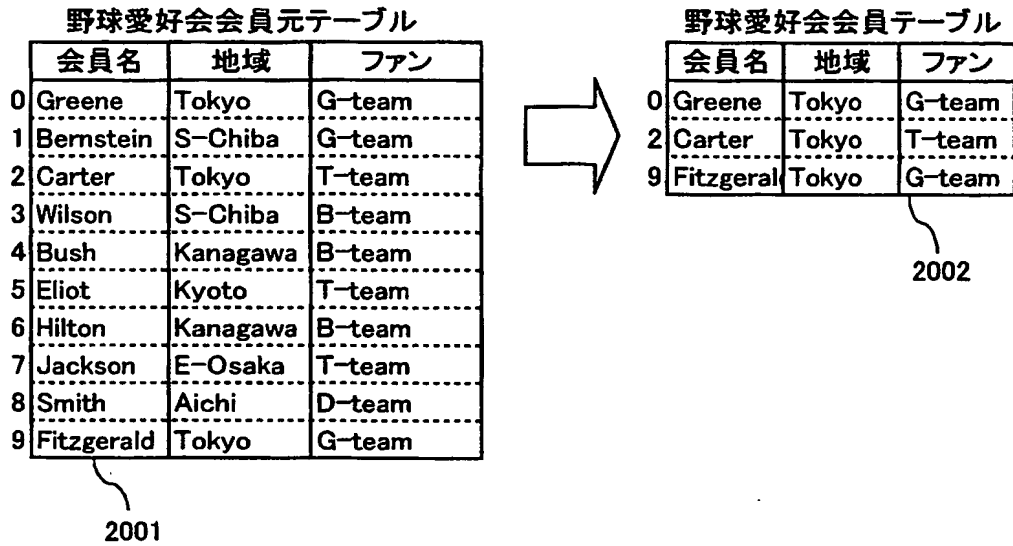
【図 19】



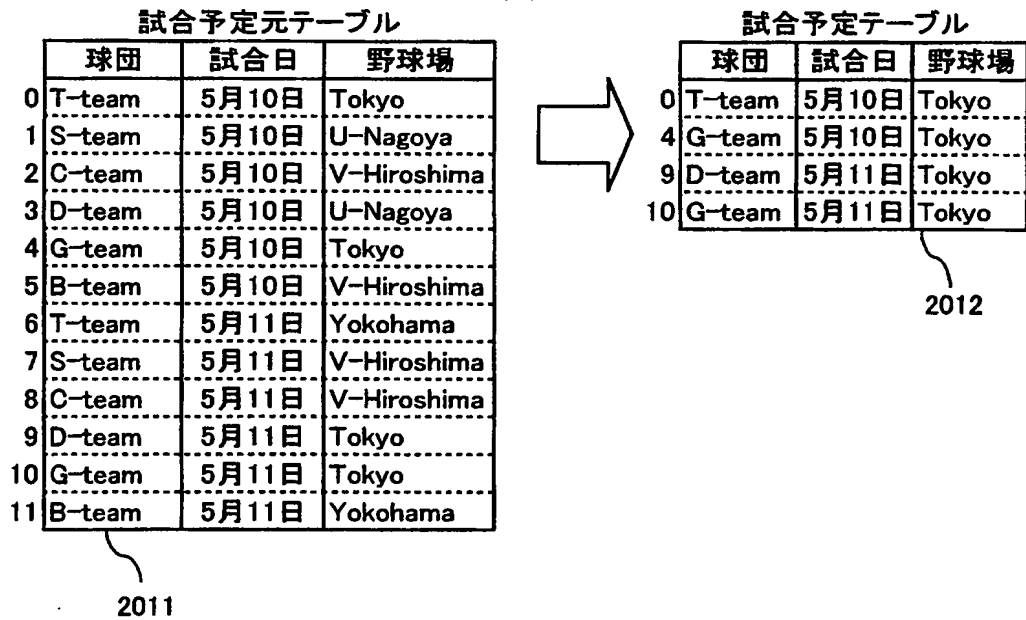
【図 20】

図20

(a)



(b)



【図 21】

図21

マスタ側			スレーブ側		
会員名	地域	ファン	球団	試合日	野球場
Greene	Tokyo	G-team	G-team	5月10日	Tokyo
Greene	Tokyo	G-team	G-team	5月11日	Tokyo
Carter	Tokyo	T-team	T-team	5月10日	Tokyo
Fitzgerald	Tokyo	G-team	G-team	5月10日	Tokyo
Fitzgerald	Tokyo	G-team	G-team	5月11日	Tokyo

2101

2111



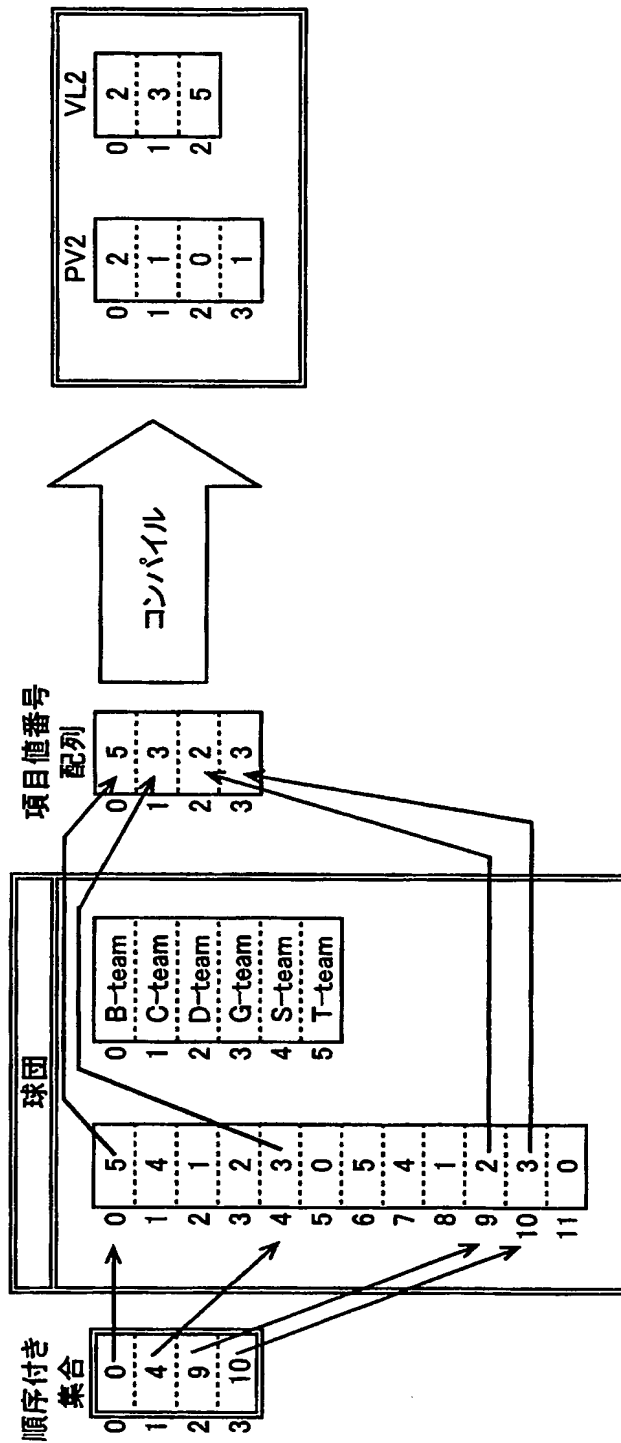
【図 22】

球団		試合日		野球場	
PV	VL	PV	VL	PV	VL
0 5	0 B-team	0 0	0 5月10日	0 0	0 Tokyo
1 4	1 C-team	1 0	1 5月11日	1 1	1 U-Nagoya
2 1	2 D-team	2 0		2 2	2 V-Hiroshima
3 2	3 G-team	3 0		3 1	3 Yokohama
4 3	4 S-team	4 0		4 0	
5 0	5 T-team	5 0		5 2	
6 5		6 1		6 3	
7 4		7 1		7 2	
8 1		8 1		8 2	
9 2		9 1		9 0	
10 3		10 1		10 0	
11 0		11 1		11 3	

図 22

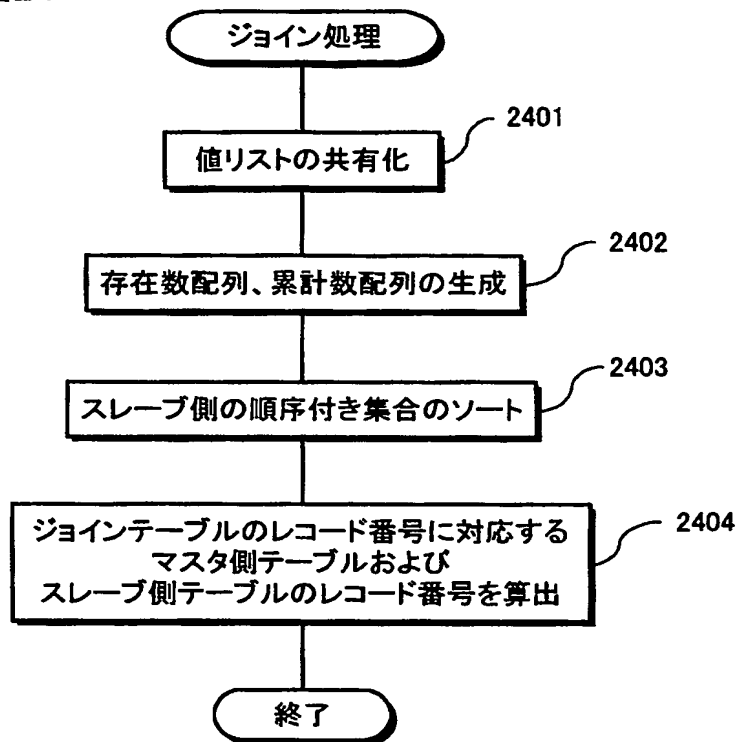
【図 23】

図23



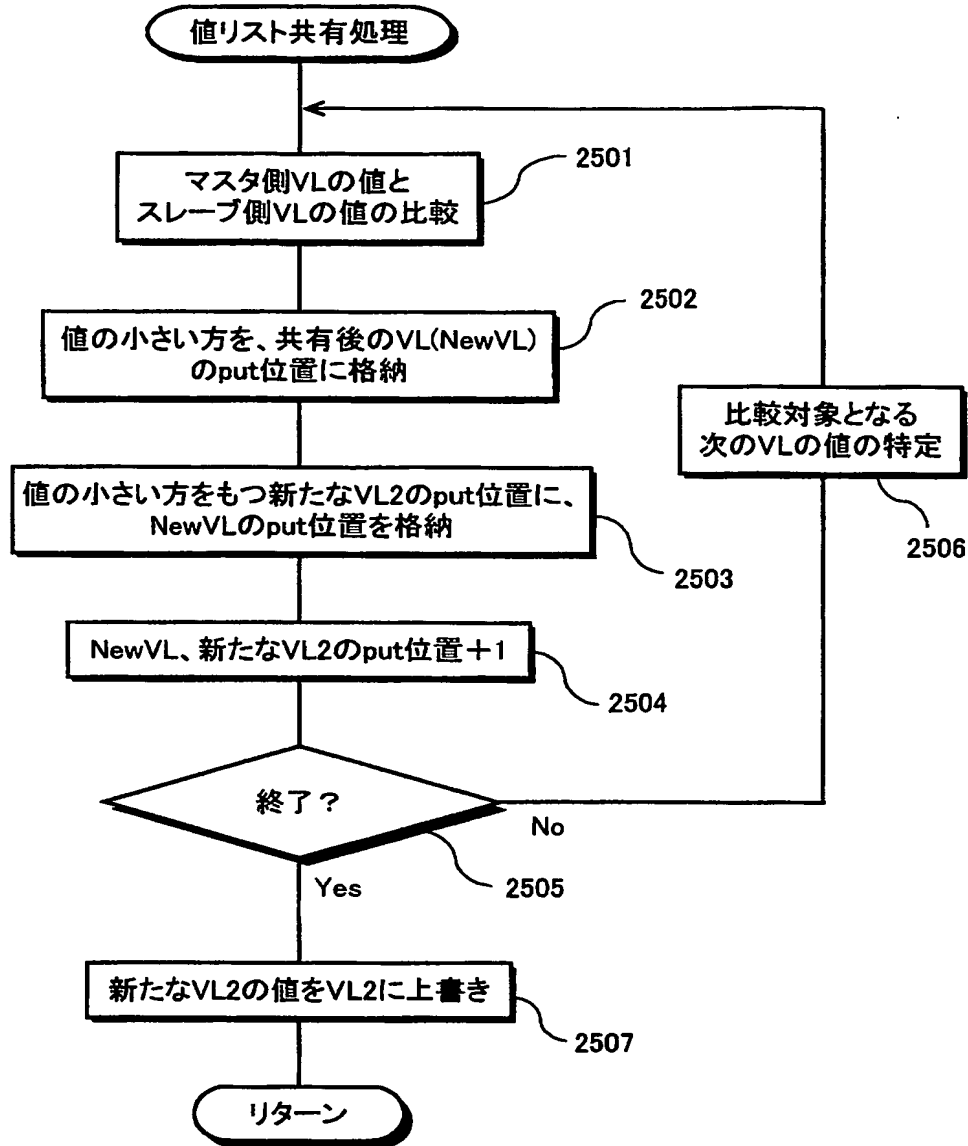
【図 24】

図24

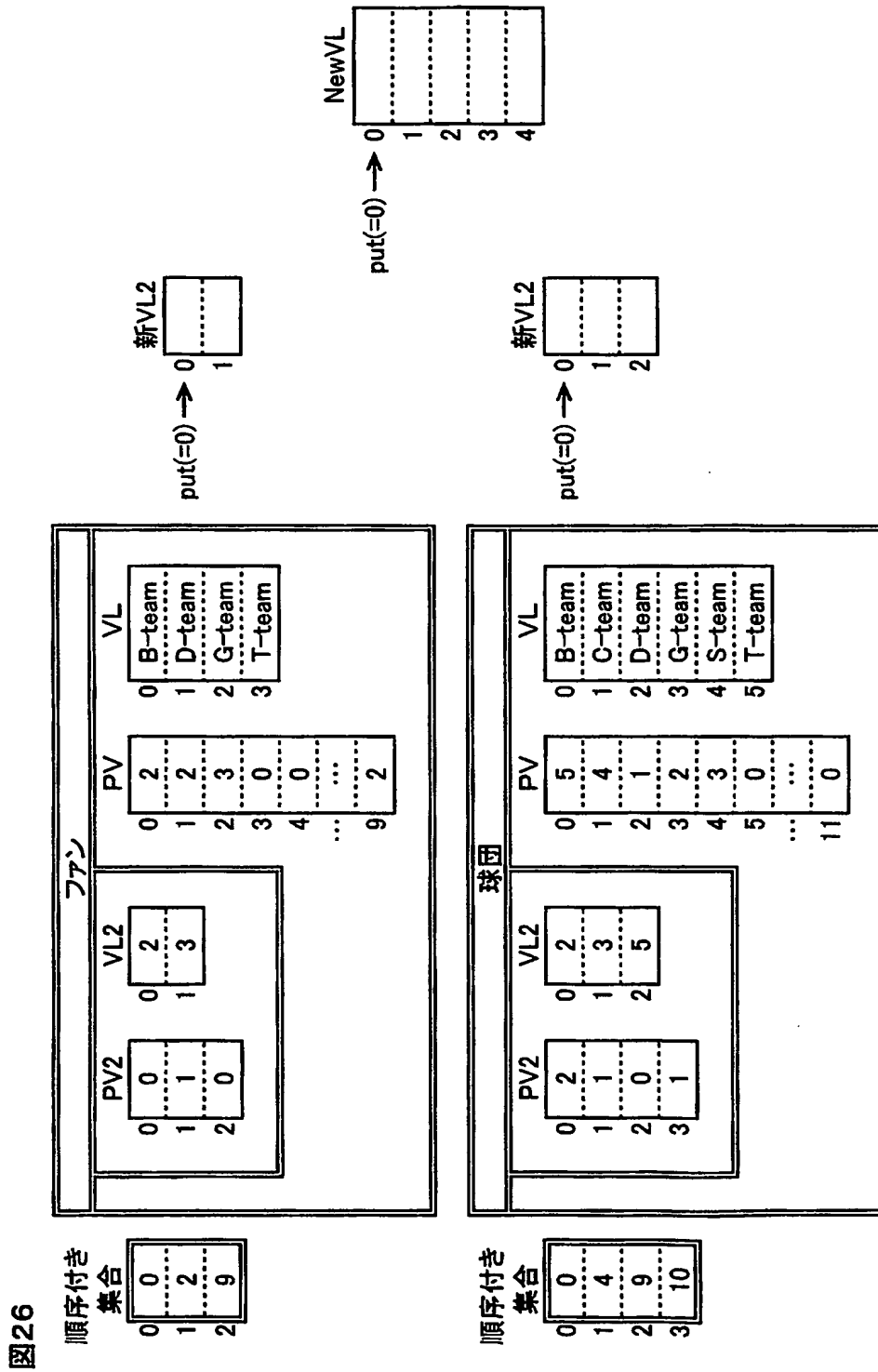


【図 25】

図25

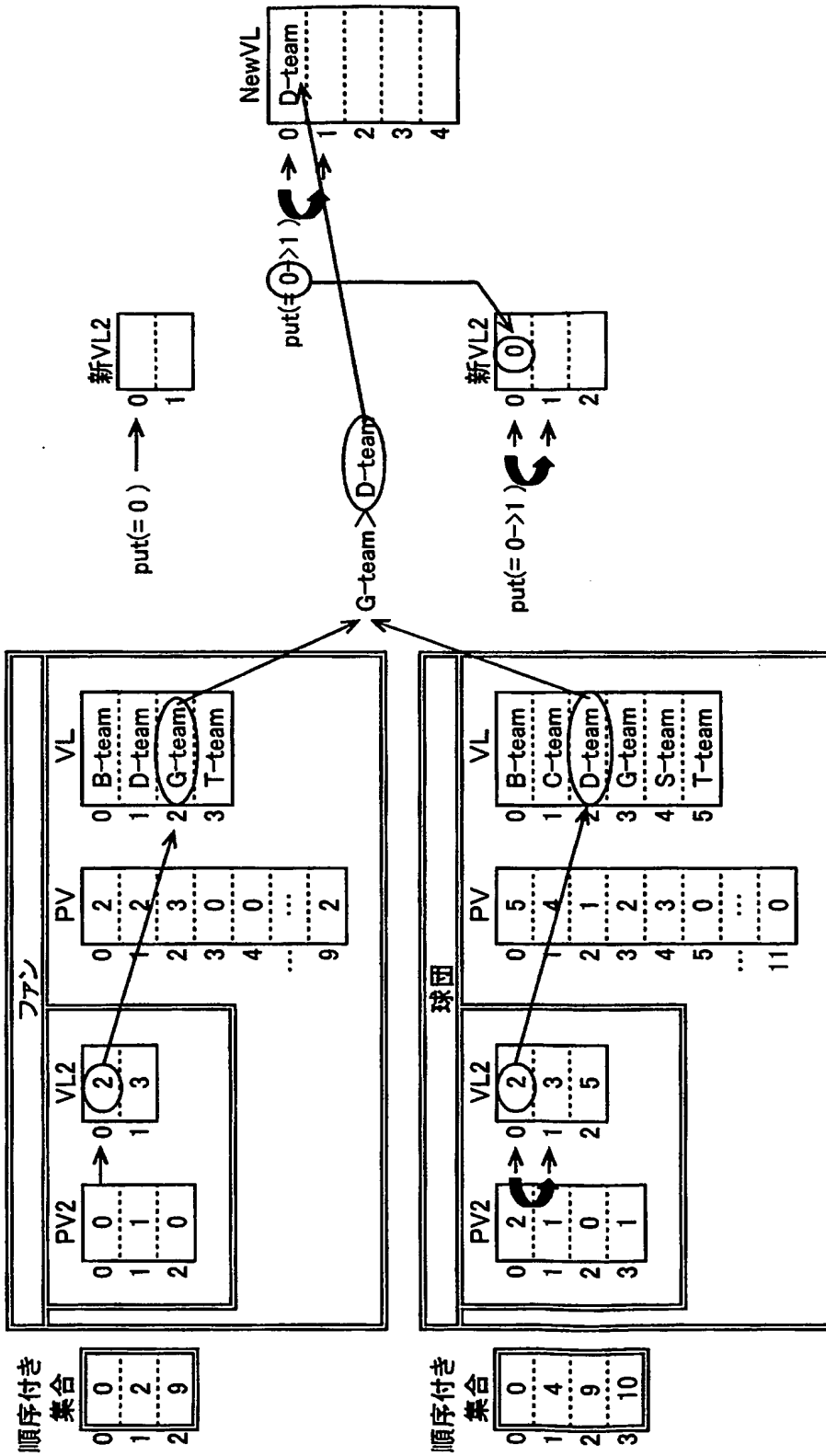


【図 26】

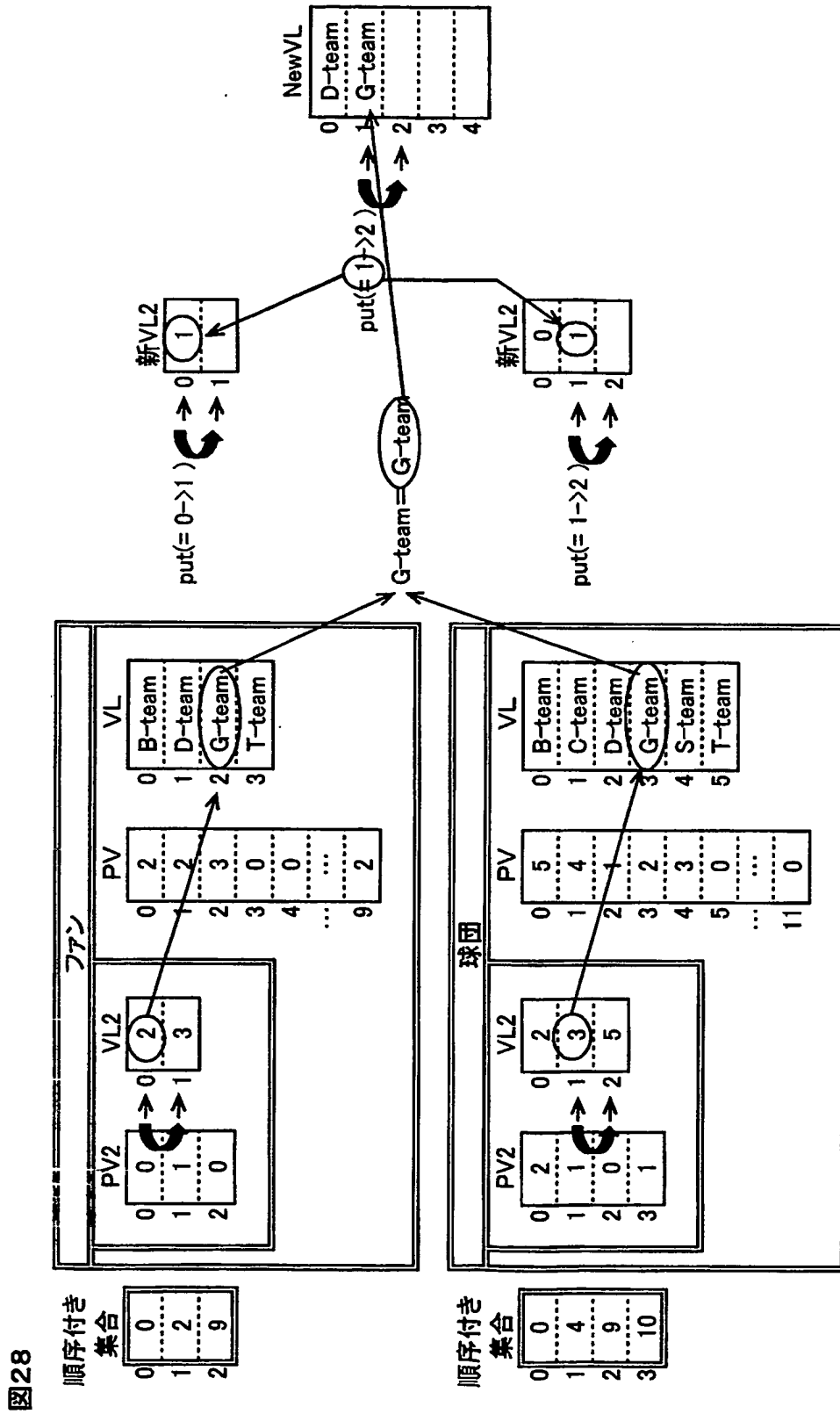


【図 27】

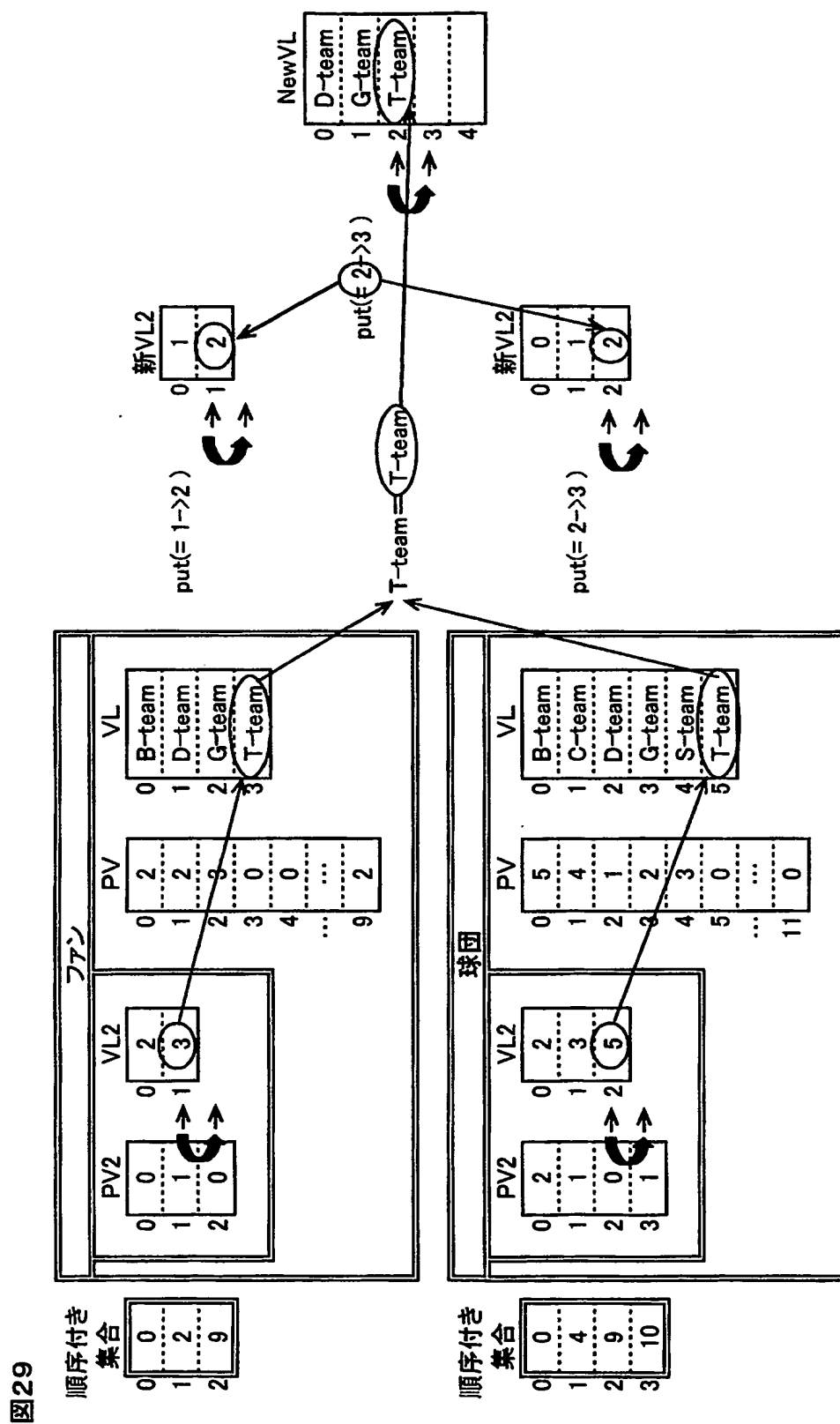
図 27



【図 28】

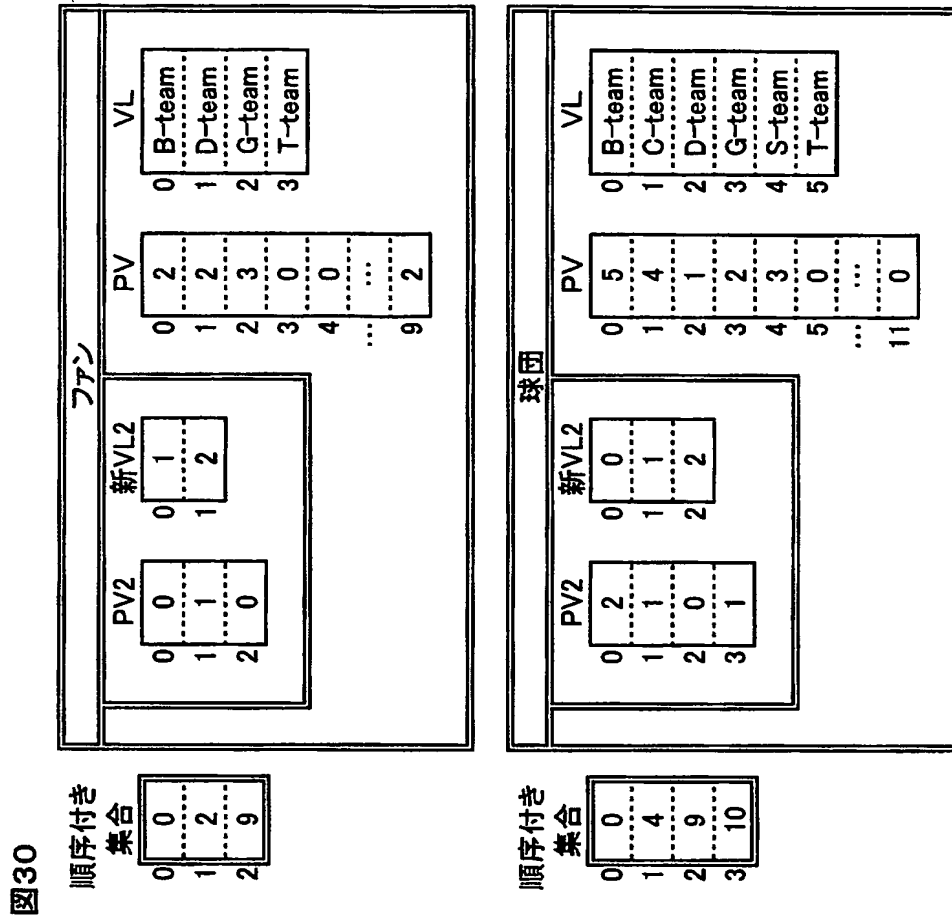


【図 29】



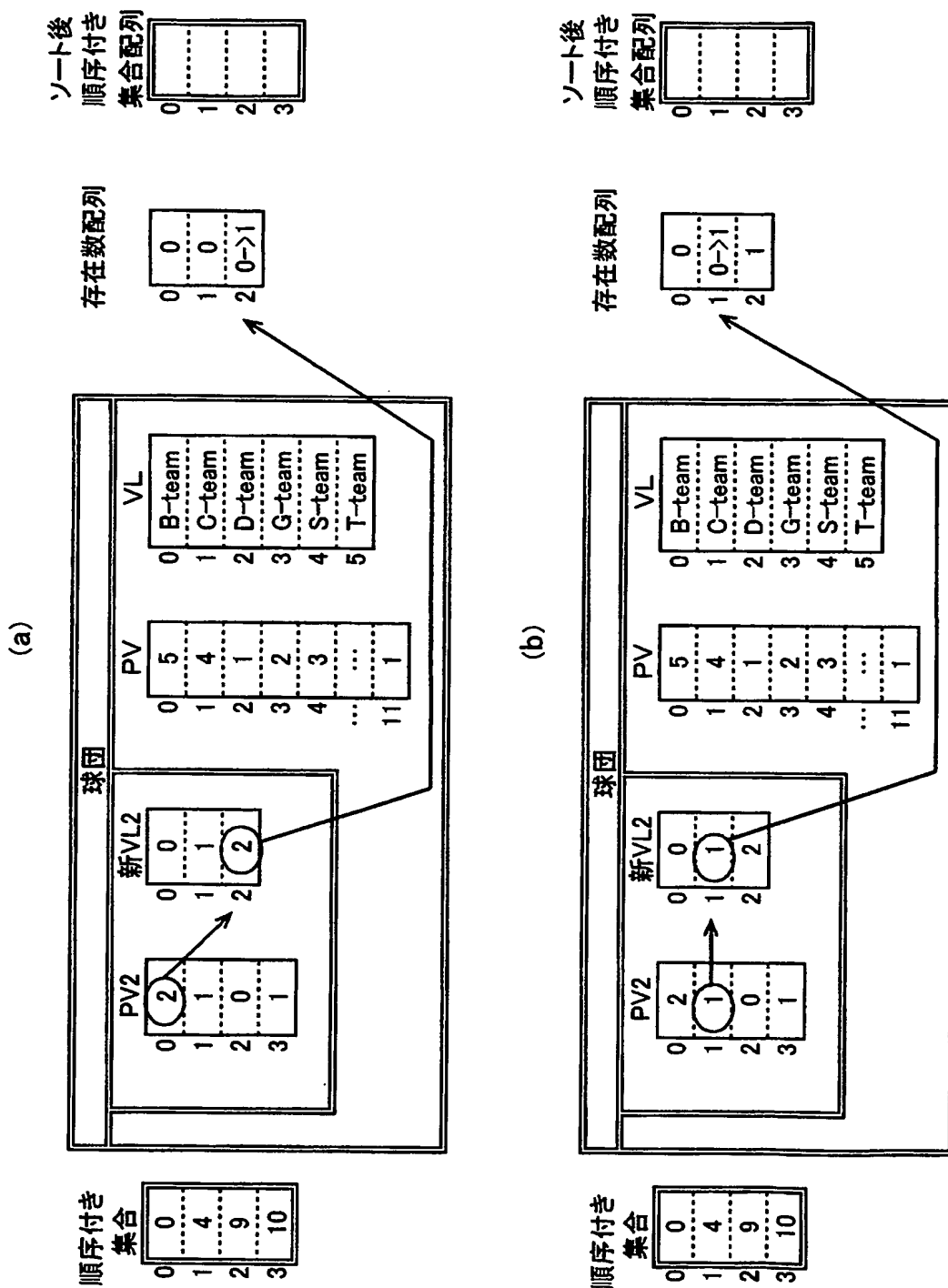


【図 30】



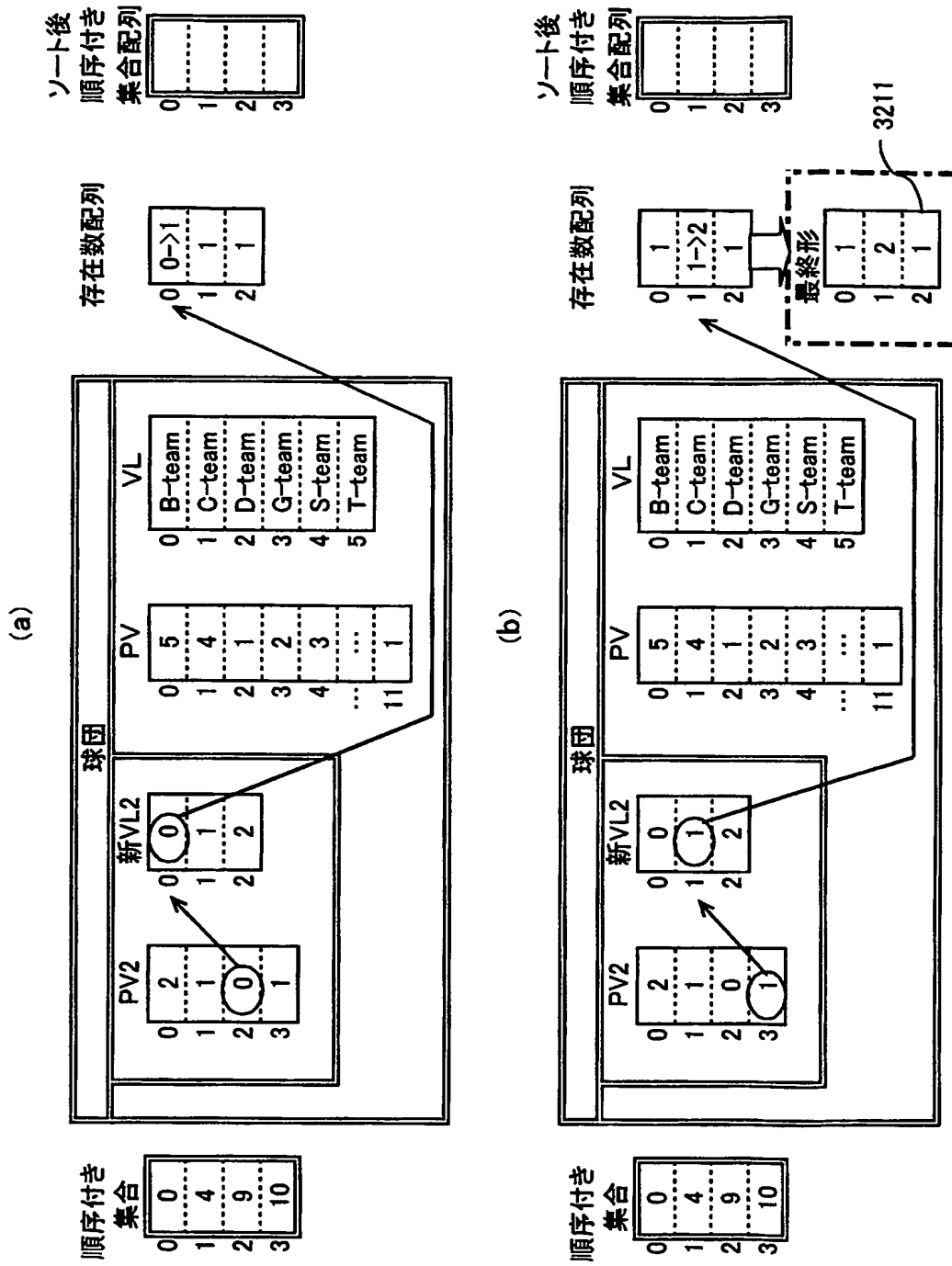
【図 31】

図31



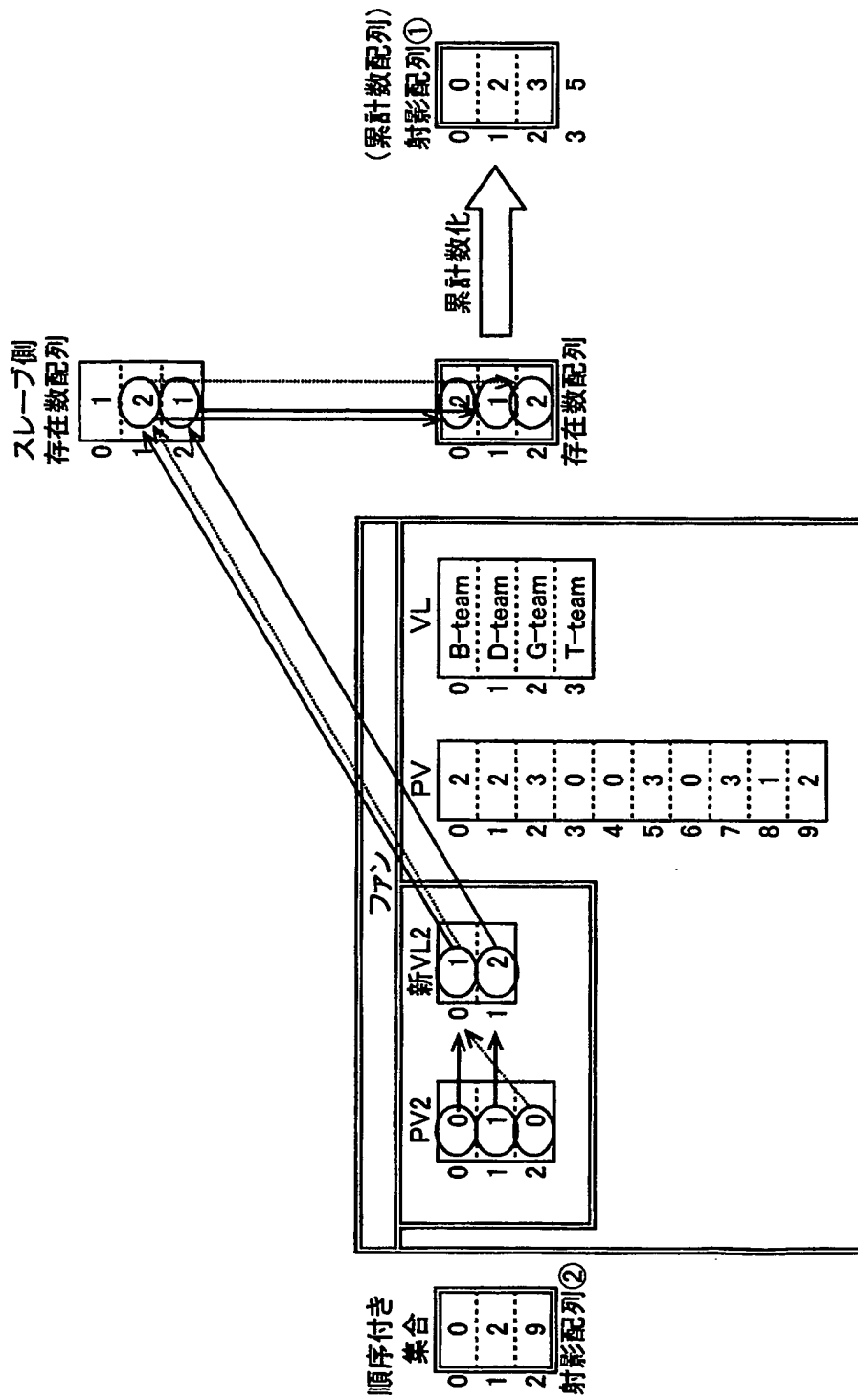
【図32】

図32

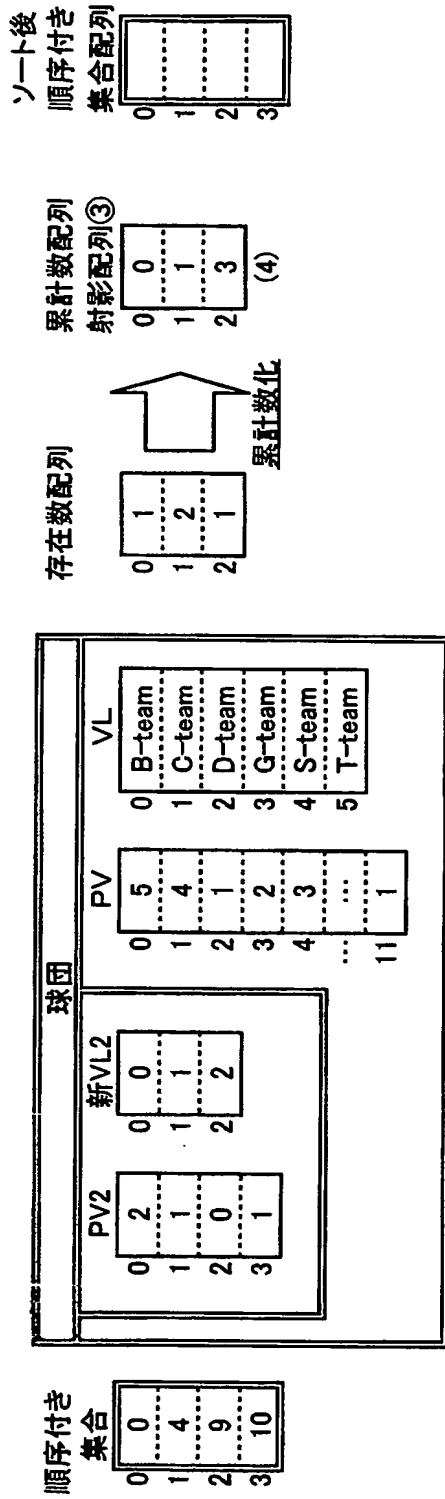


【図 33】

図33

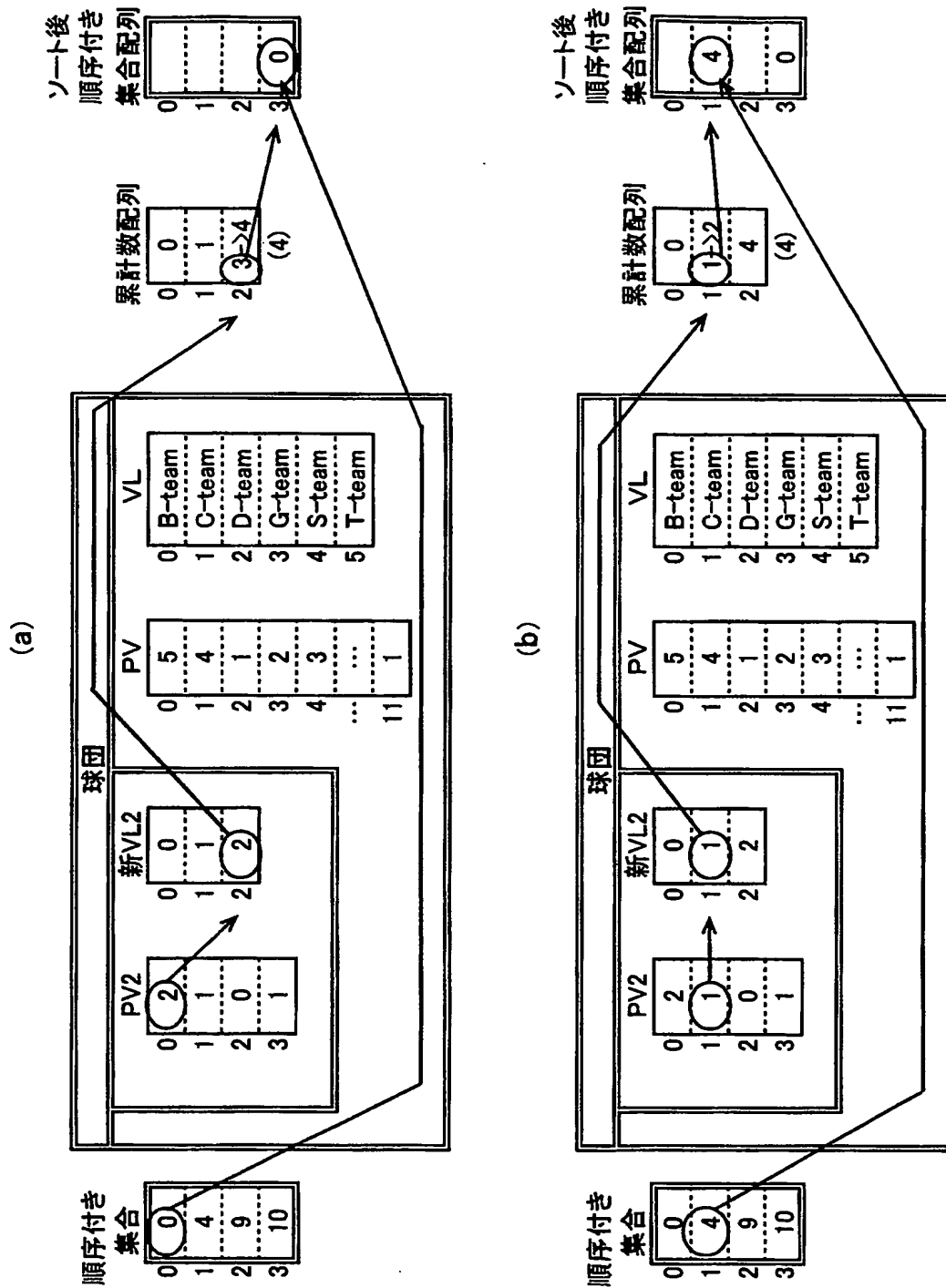


【図 34】



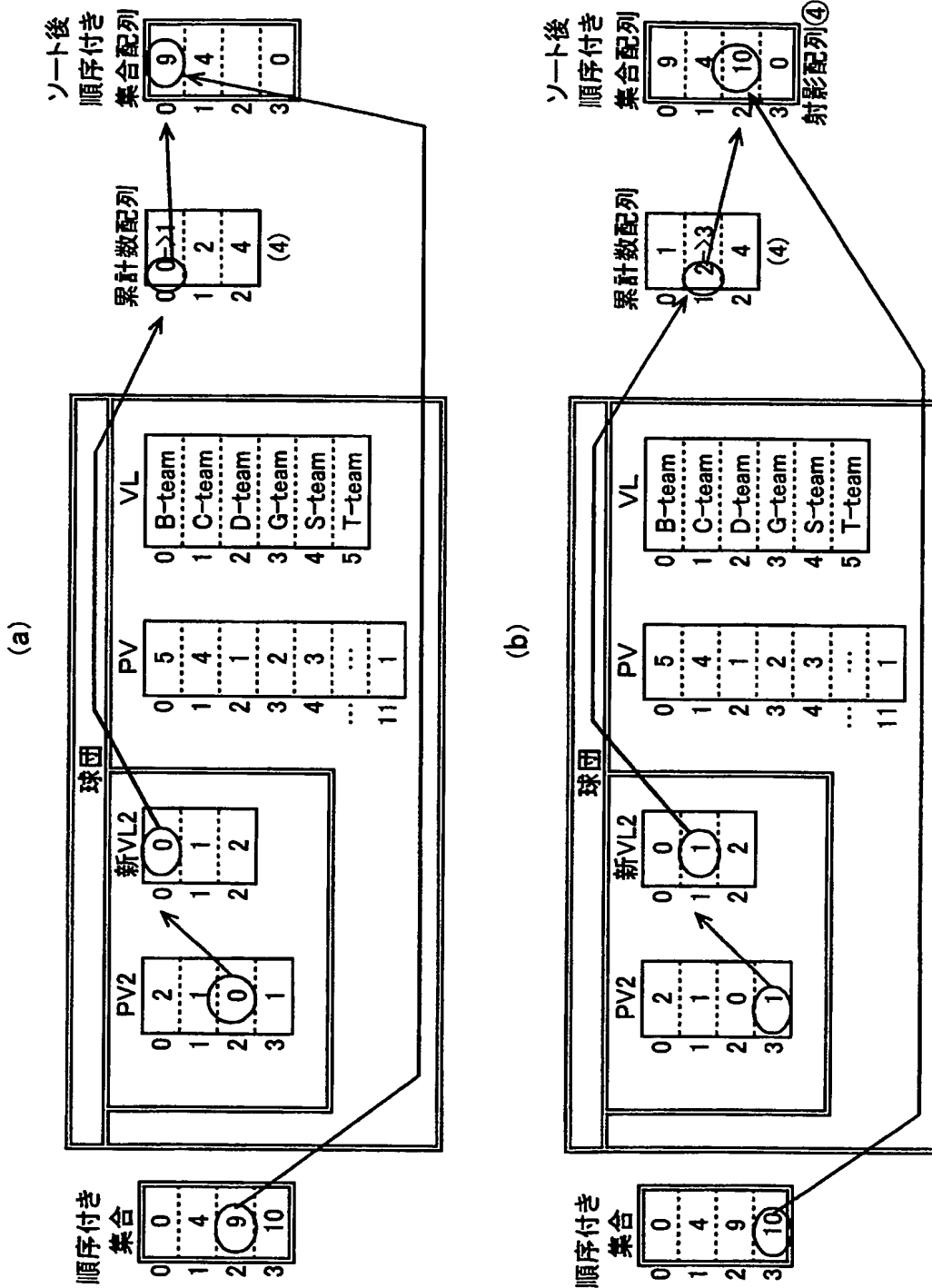
【図 35】

図 35



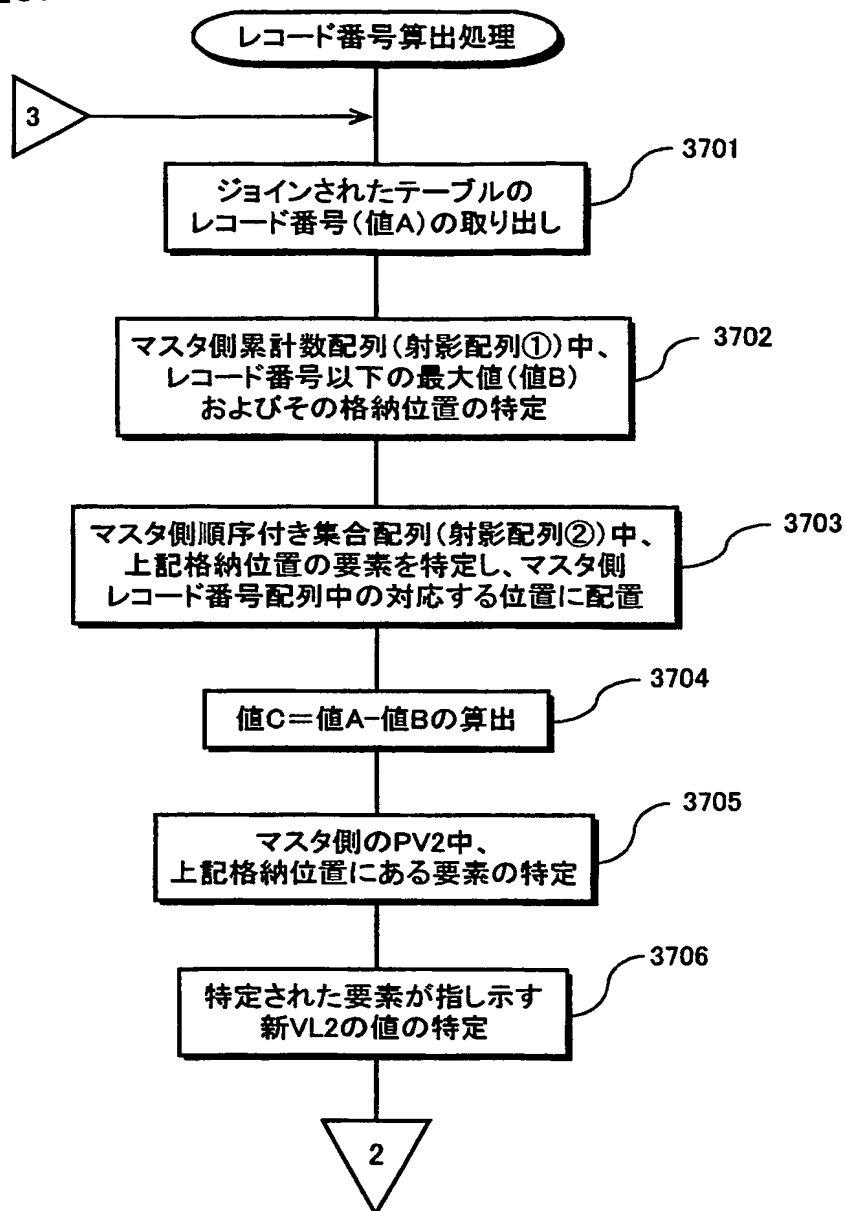
【図 36】

図 36



【図 37】

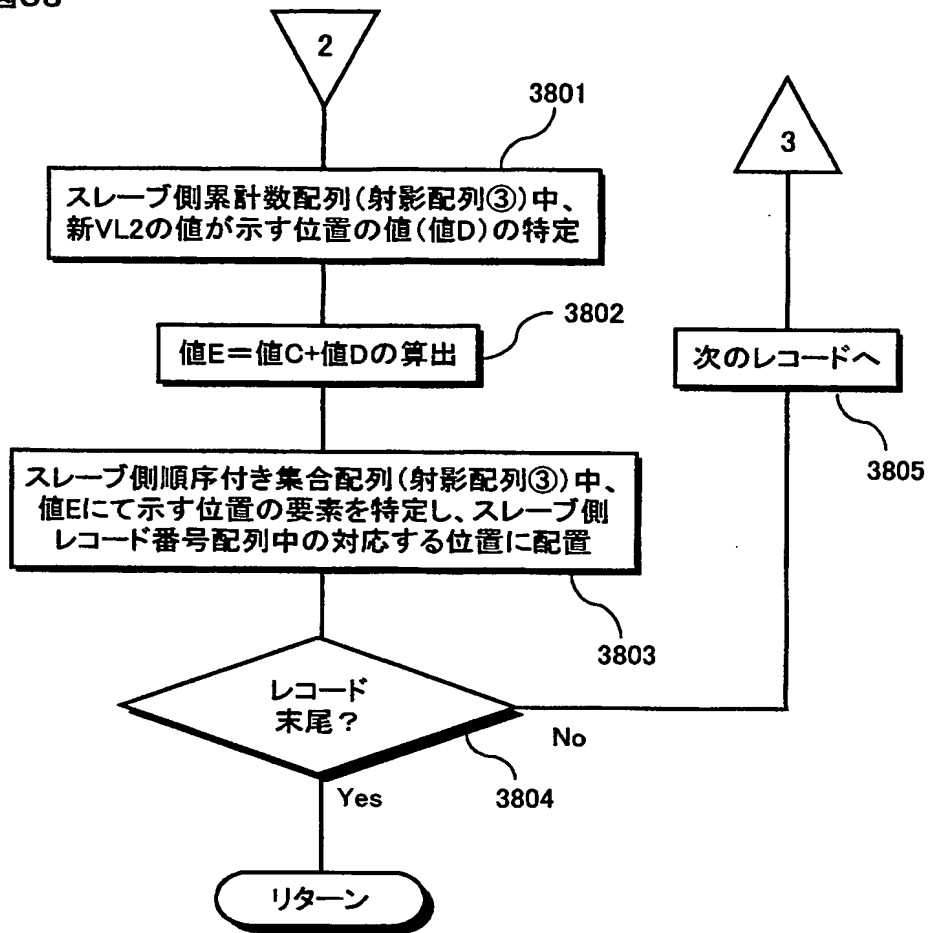
図37





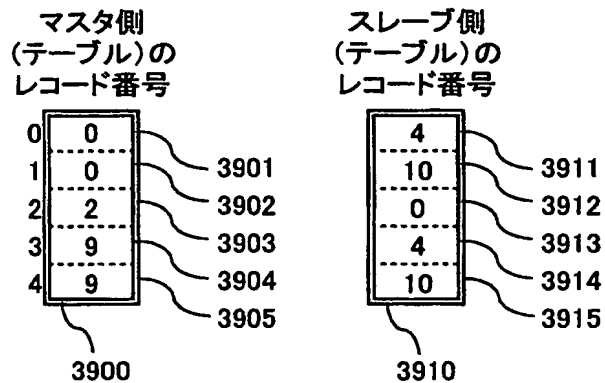
【図 38】

図38

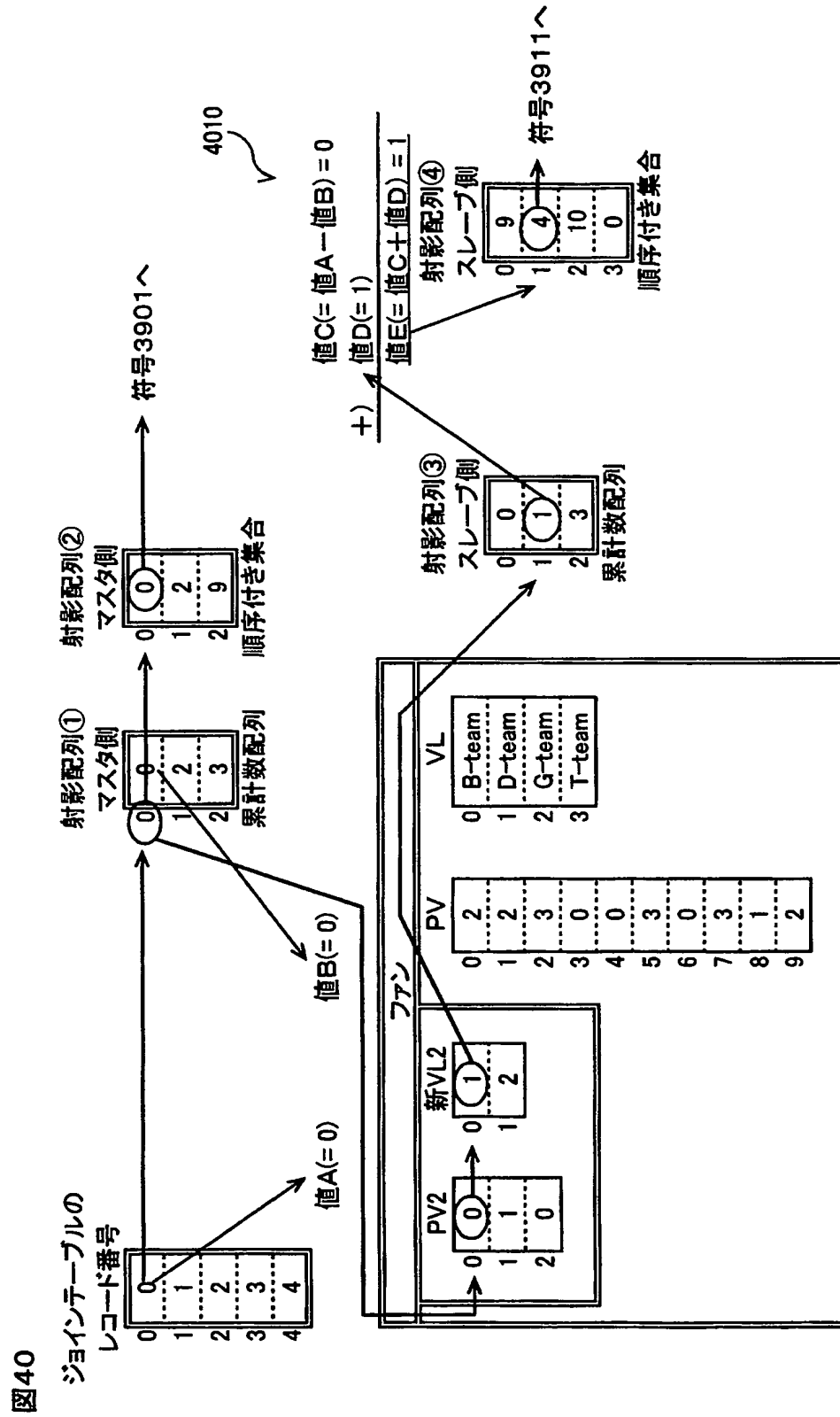


【図 39】

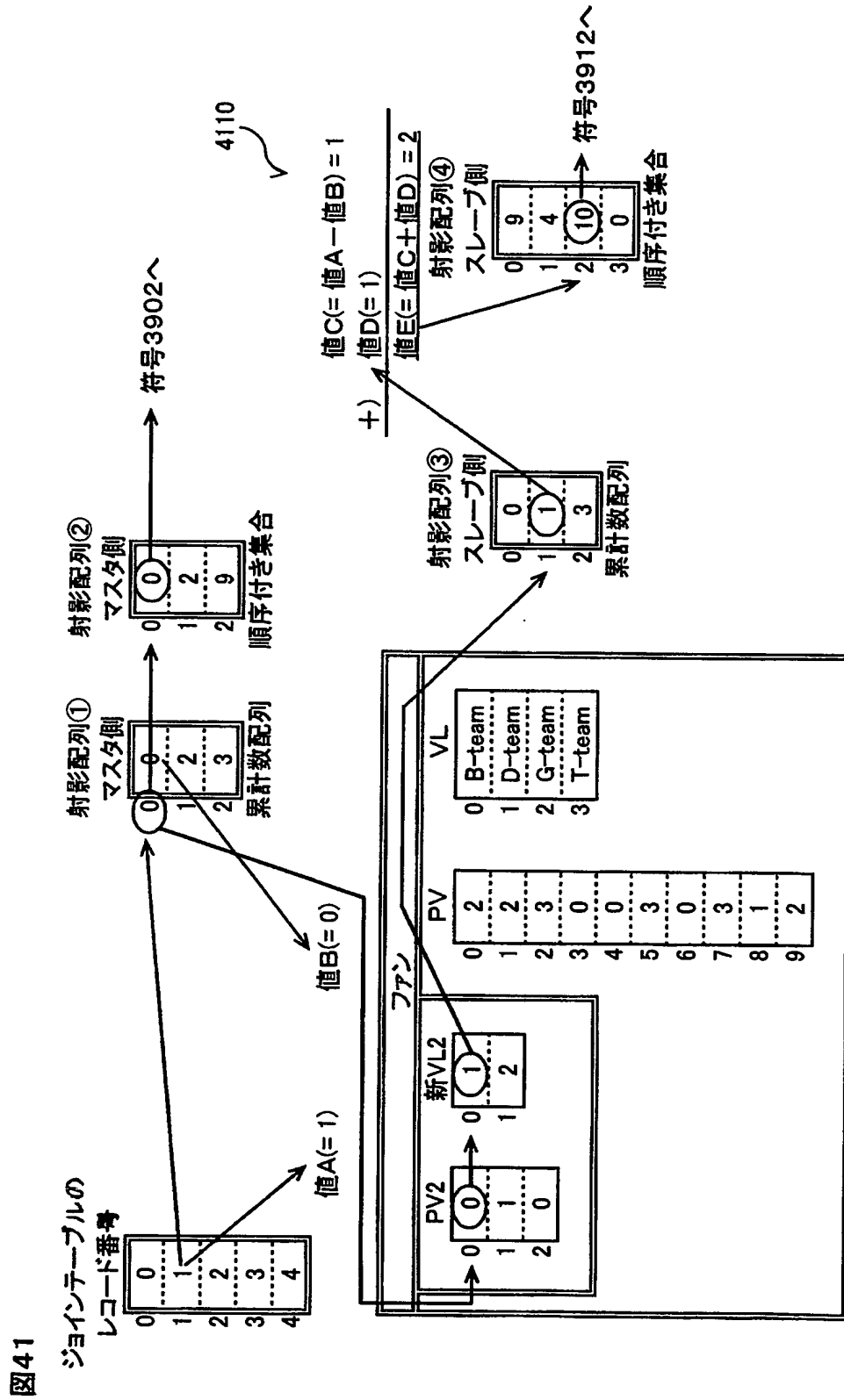
図39



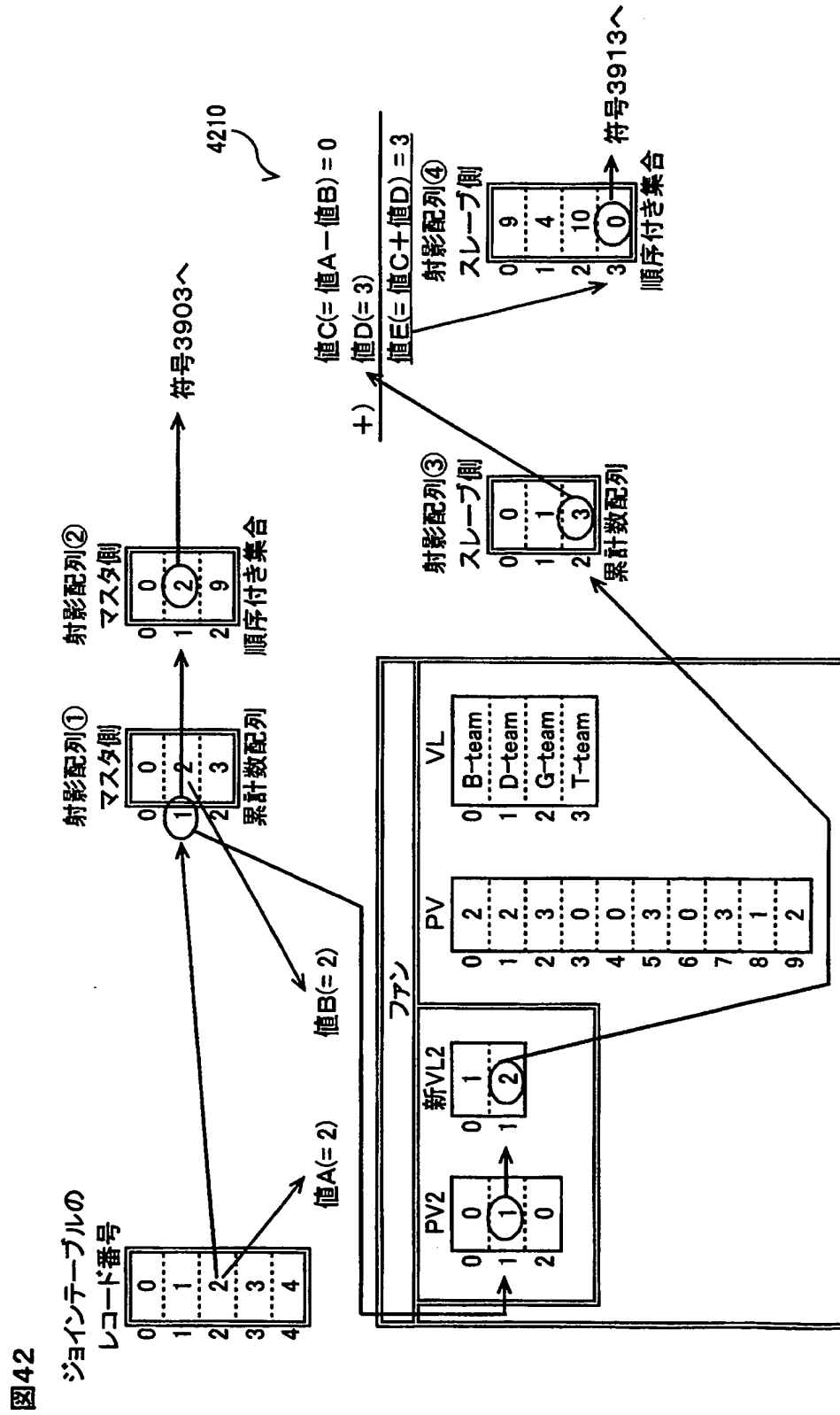
【図 40】



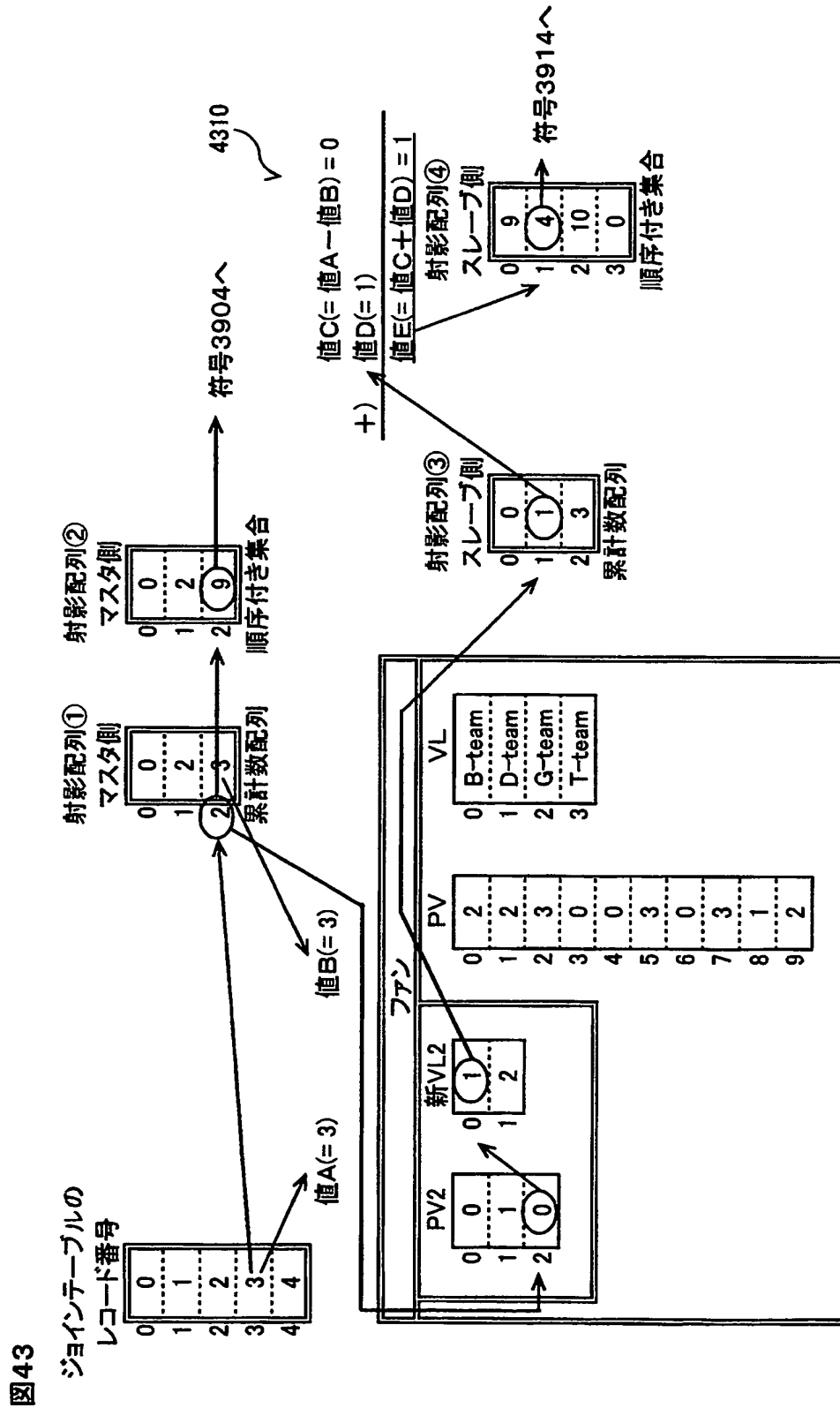
【図 41】



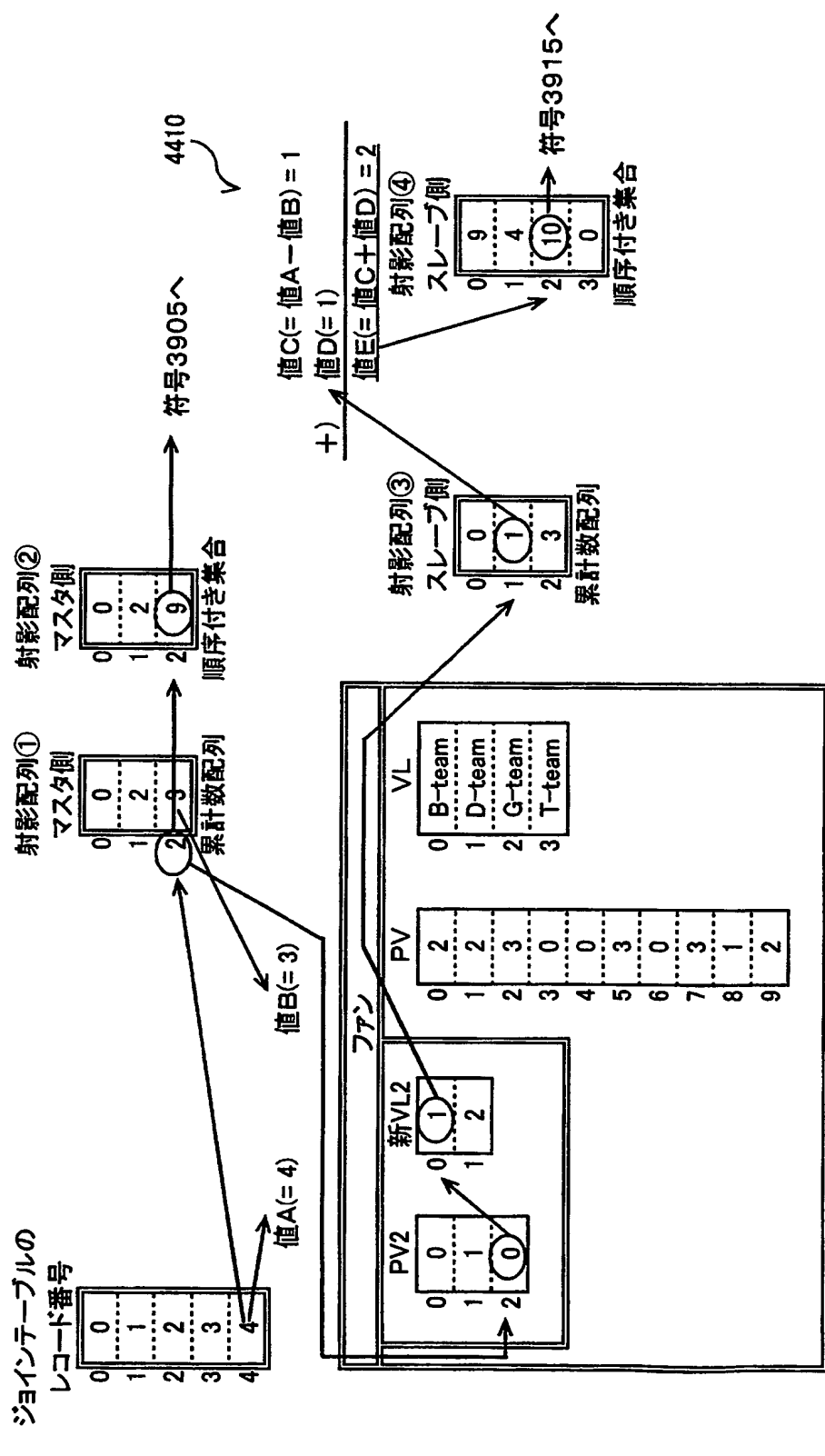
【図 4 2】



【図 43】

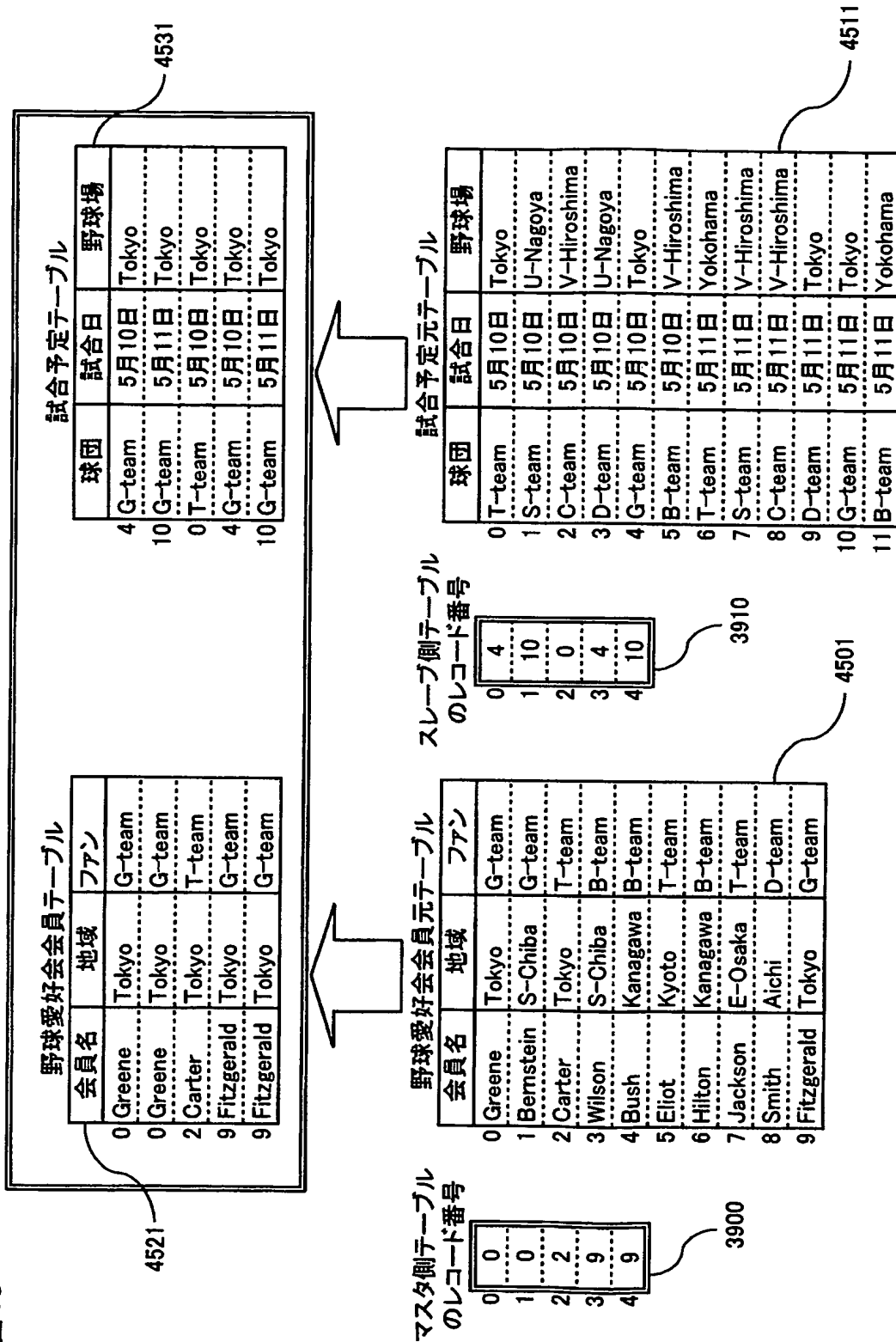


【図 4 4】



【図 45】

図 45



【書類名】 要約書

【要約】

【課題】 配列を部分集合に関する処理時間を短縮する。

【解決手段】 表形式データは、各々が、項目に属する項目値に対応した項目値番号の順に当該項目値が格納されている値リストVLと、一意的なレコード番号の順に、項目値番号を指示するためのポインタ値が格納されたポインタ配列PVとからなる情報ブロックに分割されている。レコードのうち、その一部が部分集合として選択されたことに応答して、部分集合のレコード番号を収容した順序付き集合配列が生成され、順序付き集合のそれぞれのレコード番号が指し示す位置の、前記ポインタ配列中のポインタ値が、前記レコード番号の配置された位置に対応して、項目値番号配列中に配置され、項目値番号配列中の値を参照して、位置の要素を格納した第2のポインタ配列PV2および値の要素を格納した第2の値リストVL2が生成される。

【選択図】 図10



認定・付加情報

特許出願の番号	特願 2002-310618
受付番号	50201609520
書類名	特許願
担当官	第七担当上席 0096
作成日	平成14年11月14日

<認定情報・付加情報>

【提出日】	平成14年10月25日
-------	-------------

次頁無

特願 2 0 0 2 - 3 1 0 6 1 8

出 願 人 履 歴 情 報

識別番号

[ 5 0 2 3 6 9 0 1 2 ]

1. 変更年月日

2 0 0 2 年 9 月 5 日

[変更理由]

新規登録

住 所

神奈川県横浜市神奈川区松見町四丁目 1 1 0 1 番地 7

氏 名

株式会社ターボデータラボラトリー

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**